日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: ２０００年１０月２３日

出 願 番 号
Application Number: 特願２０００－３２３１７８

出 願 人
Applicant (s): 株式会社日立製作所
株式会社日立超エル・エス・アイ・システムズ

CERTIFIED COPY OF
PRIORITY DOCUMENT

２００１年 １月１２日

特 許 庁 長 官
Commissioner,
Patent Office 及川耕造

出証番号 出証特２０００－３１１１６６９

【書類名】　特許願

【整理番号】　H00014211

【提出日】　平成12年10月23日

【あて先】　特許庁長官殿

【国際特許分類】　G06F 12/14

【発明者】

【住所又は居所】　東京都小平市上水本町五丁目２０番１号　株式会社　日立製作所　半導体グループ内

【氏名】　谷本　千晶

【発明者】

【住所又は居所】　東京都小平市上水本町五丁目２０番１号　株式会社　日立製作所　半導体グループ内

【氏名】　中田　邦彦

【発明者】

【住所又は居所】　東京都小平市上水本町５丁目２２番１号　日立超エル・エス・アイ・システムズ内

【氏名】　成吉　雄一郎

【発明者】

【住所又は居所】　東京都小平市上水本町５丁目２２番１号　日立超エル・エス・アイ・システムズ内

【氏名】　塚元　卓

【発明者】

【住所又は居所】　東京都小平市上水本町５丁目２２番１号　日立超エル・エス・アイ・システムズ内

【氏名】　平林　茂雄

【発明者】

【住所又は居所】　東京都小平市上水本町５丁目２２番１号　日立超エル・エス・アイ・システムズ内

【氏名】　渡瀬　弘

【発明者】

【住所又は居所】　東京都小平市上水本町５丁目２２番１号　日立超エル・エス・アイ・システムズ内

【氏名】　▲高▼橋　雅聡

【特許出願人】

【識別番号】　000005108

【氏名又は名称】　株式会社　日立製作所

【特許出願人】

【識別番号】　000233169

【氏名又は名称】　株式会社　日立超エル・エス・アイ・システムズ

【代理人】

【識別番号】　100081938

【弁理士】

【氏名又は名称】　徳若　光政

【電話番号】　0422-46-5761

【先の出願に基づく優先権主張】

【出願番号】　特願2000-　3297

【出願日】　平成12年 1月12日

【手数料の表示】

【予納台帳番号】　000376

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9003106

【包括委任状番号】　9107732

【プルーフの要否】　要

【書類名】　　明細書

【発明の名称】　ＩＣカードとマイクロコンピュータ

【特許請求の範囲】

【請求項１】　外部端子がリードライト装置と電気的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ったデータの入出力動作を含むＩＣカードであって、

上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうことを特徴とするＩＣカード。

【請求項２】　請求項１において、

上記暗号化処理又は復号化処理は、ＲＳＡ暗号法などに応用可能なべき乗剰余乗算動作を含むものであることを特徴とするＩＣカード。

【請求項３】　請求項２において、

上記べき乗剰余乗算動作は、中央処理装置からの指示を受けて動作する暗号処理用演算ユニットにより行なわれるものであることを特徴とするＩＣカード。

【請求項４】　請求項３において、

上記暗号化処理用演算ユニットは、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、$A=A^2 \bmod N$と$A=AB \bmod N$の演算を交互に行ない、かかる演算においてＹの上位から１ビットずつみて、論理０であれば上記$A^2 \bmod N$の演算結果を有効なデータとして記憶回路に取り込み、論理１であれば上記$A^2 \bmod N$と$AB \bmod N$の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理０のときの$A=AB \bmod N$の演算動作が上記攪乱目的のダミー処理動作とされることを特徴とするＩＣカード。

【請求項５】　請求項４において、

上記記憶回路は、リードライトバッファとかかるリードライトバッファを通してデータの入出力が行なわれる複数のレジスタとからなるレジスタブロックであり、

上記演算結果は、上記Ｙの特定ビットe_iの論理１又は０によってゲート回路を制御し、所定のレジスタに供給されるライトストローブ信号の伝達を制御して

、有効なデータのみがリードライトバッファを通して上記所定のレジスタに格納されることを特徴とするＩＣカード。

【請求項６】　請求項４において、

上記記憶回路は、リードライトバッファとかかるリードライトバッファを通してデータの入出力が行なわれる複数のレジスタとからなるレジスタブロックであり、

上記演算結果は、上記Ｙの特定ビットe_iの論理１又は０によってゲート回路を制御し、上記リードライトバッファに供給されるライトストローブ信号の伝達を制御して、有効なデータのみがリードライトバッファを通して上記所定のレジスタに格納されることを特徴とするＩＣカード。

【請求項７】　請求項４において、

上記記憶回路は、リードライトバッファとかかるリードライトバッファを通してデータの入出力が行なわれる複数のレジスタ及びダミーレジスタとからなるレジスタブロックであり、

上記演算結果は、上記リードライトバッファと上記ダミーレジスタ及び複数のレジスタとの間に設けられたセレクタを上記Ｙの特定ビットe_iの論理１又は０によって制御して上記リードライトバッファに書き込まれた演算結果のうち有効なデータが所定のレジスタに格納され、無効なデータが上記ダミーレジスタに格納されるものであることを特徴とするＩＣカード。

【請求項８】　請求項３において、

上記暗号化処理用演算ユニットは、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、$A=A^2 \bmod N$と$A=AB \bmod N$の演算を交互に行ない、かかる演算においてＹの上位から１ビットずつみて、論理０であれば上記$A^2 \bmod N$の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込み、論理１であれば上記$A^2 \bmod N$と$AB \bmod N$の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込むものであり、上記$A=A^2 \bmod N$の演算結果が出力されてから上記$A=AB \bmod N$の演算が開始されるまでの間も上記$A=A^2 \bmod N$の動作を継続し、$A=AB \bmod N$の演算結果が出力されてからＹのビットの変更判定処理を含めて次のビットに対応した$A^2 \bmod$

Ｎの演算が開始されるまでの間も上記Ａ＝ＡＢｍｏｄＮの動作を継続するものであることを特徴とするＩＣカード。

【請求項９】　請求項３において、

上記暗号化処理用演算ユニットは、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、$A=A^2 \bmod N$とＡ＝ＡＢｍｏｄＮの演算とそれぞれに対してオーバーフロー演算行ない、かかる演算においてＹの上位から１ビットずつみて、論理０であれば上記$A^2 \bmod N$の演算結果を有効なデータとして記憶回路に取り込み、論理１であれば上記$A^2 \bmod N$とＡＢｍｏｄＮの演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理０のときのＡ＝ＡＢｍｏｄＮの演算動作と、各演算動作での不要なオーバーフロー演算が上記攪乱目的のダミー処理動作とされることを特徴とするＩＣカード。

【請求項１０】　外部端子がリードライト装置と電気的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ってデータの入出力動作が行われるＩＣカードであって、

上記暗号化処理又は復号化処理に攪乱目的のダミー演算を含ませて内部回路の動作タイミング及び動作電流に不規則性を持たせてなることを特徴とするＩＣカード。

【請求項１１】　外部端子がリードライト装置と電気的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ってデータの入出力動作が行われるＩＣカードであって、

上記暗号化処理又は復号化処理における各演算の間隔に攪乱目的のダミーサイクルを含ませて内部回路の動作タイミング及び動作電流に不規則性を持たせてなることを特徴とするＩＣカード。

【請求項１２】　暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータであって、

上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうことを特徴とするマイクロコンピュータ。

【請求項１３】　請求項１２において、

上記モジュール構成は、１つの半導体基板上において形成されることによって実現されることを特徴とするマイクロコンピュータ。

【請求項１４】　請求項１３において、

上記暗号化処理又は復号化処理は、ＲＳＡ暗号法などに応用可能なべき乗剰余乗算動作を含み、

上記べき乗剰余乗算動作は、中央処理装置からの指示を受けて動作する暗号処理用演算ユニットにより行なわれるものであることを特徴とするマイクロコンピュータ。

【請求項１５】　請求項１４において、

上記暗号化処理用演算ユニットは、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、$A=A^2 \bmod N$と$A=AB \bmod N$の演算を交互に行ない、かかる演算においてＹの上位から１ビットずつみて、論理０であれば上記$A^2 \bmod N$の演算結果を有効なデータとして記憶回路に取り込み、論理１であれば上記$A^2 \bmod N$と$AB \bmod N$の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理０のときの$A=AB \bmod N$の演算動作が上記攪乱目的のダミー処理動作とされることを特徴とするマイクロコンピュータ。

【請求項１６】　請求項１４において、

上記暗号化処理用演算ユニットは、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、$A=A^2 \bmod N$と$A=AB \bmod N$の演算を交互に行ない、かかる演算においてＹの上位から１ビットずつみて、論理０であれば上記$A^2 \bmod N$の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込み、論理１であれば上記$A^2 \bmod N$と$AB \bmod N$の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込むものであり、上記$A=A^2 \bmod N$の演算結果が出力されてから上記$A=AB \bmod N$の演算が開始されまるまでの間も上記$A=A^2 \bmod N$の動作を継続し、$A=AB \bmod N$の演算結果が出力されてからＹのビットの変更判定処理を含めて次のビットに対応した$A^2 \bmod N$の演算が開始されるまでの間も上記$A=AB \bmod N$の動作を継続するものであることを特徴とするマイクロコンピュータ。

【請求項１７】　請求項１４において、

上記暗号化処理用演算ユニットは、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、$A=A^2 \bmod N$と$A=AB \bmod N$の演算とそれぞれに対してオーバーフロー演算行ない、かかる演算においてＹの上位から１ビットずつみて、論理０であれば上記$A^2 \bmod N$の演算結果を有効なデータとして記憶回路に取り込み、論理１であれば上記$A^2 \bmod N$と$AB \bmod N$の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理０のときの$A=AB \bmod N$の演算動作と、各演算動作での不要なオーバーフロー演算が上記攪乱目的のダミー処理動作とされることを特徴とするマイクロコンピュータ。

【請求項１８】　請求項３において、

上記暗号化処理用演算ユニットは、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、Ｙのビットの値に応じて、$A=A^2 R^{-1} \bmod N$、$A=ABR^{-1} \bmod N$の演算を行うとともに、

演算結果にオーバーフローが発生した場合にはさらに上記演算結果ＷからＮの減算Ｗ－Ｎを行なう正規動作と、各々の演算結果にオーバーフローが発生しない場合でも上記減算Ｗ－Ｎに対応した演算による無効データを生成する攪乱目的のダミー動作を行い、

上記オーバーフローの有無に対応して有効なデータを出力させることを特徴とするＩＣカード。

【請求項１９】　請求項１８において、

上記$A^2 R^{-1} \bmod N$又は$ABR^{-1} \bmod N$の演算結果Ｗは第１の記憶回路に格納され、

演算器のオーバーフローフラグＯＶの有無が記憶され、

上記剰余乗算の後に上記第１記憶回路の演算結果ＷからＮの減算Ｗ－Ｎが行われて、その演算結果が上記オーバーフローフラグＯＶが有る時には上記第１の記憶回路に格納され、オーバーフローフラグＯＶが無い時には上記第１記憶回路とは異なる第２の記憶回路に上記錯乱目的のダミー動作として書き込まれ、

上記第１の記憶回路の演算結果が有効なデータとして出力されることを特徴とするＩＣカード。

【請求項２０】　請求項１８において、

上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗは第１の記憶回路に格納され、

演算器のオーバーフローフラグＯＶの有無が記憶され、

上記剰余乗算の後に上記第１の記憶回路の演算結果ＷからＮの減算Ｗ－Ｎが行われて、オーバーフローフラグＯＶが有るときに上記演算結果Ｗ－Ｎがセレクタにより選択され、オーバーフローフラグＯＶが無いときには上記第１記憶回路の演算結果Ｗがセレクタにより選択されて第２の記憶回路に格納されて有効なデータとして出力されること特徴とするＩＣカード。

【請求項２１】　請求項１８において、

上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗは第１の記憶回路に格納され、

演算器のオーバーフローフラグＯＶの有無が記憶され、

上記剰余乗算の後に上記第１の記憶回路の演算結果ＷからＮの減算Ｗ－Ｎが行われ、オーバーフローフラグＯＶが有るときには減算Ｗ－Ｎが第２の記憶回路に記憶され、オーバーフローフラグＯＶが無いときには減算Ｗ－Ｎが第３の記憶回路に記憶され、

オーバーフローフラグＯＶが有るときには上記第２の記憶回路のデータが有効なデータとして出力され、

オーバーフローフラグＯＶが無いときには上記第１の記憶回路のデータが有効なデータとして出力されること特徴とするＩＣカード。

【請求項２２】　請求項１８において、

上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗは第１の記憶回路に格納され、

演算器のオーバーフローフラグＯＶの有無が記憶され、

上記剰余乗算の後に上記第１の記憶回路の演算結果ＷからＮの減算結果Ｗ－Ｎが第２の記憶回路に格納され、オーバーフローフラグＯＶが無いとき第１の記憶回路と第２の記憶回路を選択する最下位アドレスを逆転させて、上記第２の記憶回路を選択するアドレスにより第１の記憶回路を選択して有効なデータとして出力させ、オーバーフローフラグＯＶが有るとき第１の記憶回路と第２の記憶回路

を選択する最下位アドレスをそのままにして第２の記憶回路の演算結果を有効なデータとして出力させることを特徴とするＩＣカード。

【請求項２３】　請求項１８において、

上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗは第１の記憶回路に格納され、

演算器のオーバーフローフラグＯＶの有無が記憶され、

上記剰余乗算の後に上記第１の記憶回路と第２の記憶回路のアドレスが交換され、第２の記憶回路を選択するアドレスにより選択される演算結果値ＷからＮの減算Ｗ－Ｎが行われて第１の記憶回路を選択するアドレスにより選択される第２の記憶回路に減算結果Ｗ－Ｎが格納され、オーバーフローフラグＯＶが有るときにのみ上記アドレスを再度交換し、第１の記憶回路を選択するアドレスにより選択される第１又は第２の記憶回路のデータを有効なデータとして出力させることを特徴とするＩＣカード。

【請求項２４】　請求項１８において、

上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗは第１の記憶回路に格納され、

上記剰余乗算の後に上記第１の記憶回路の演算結果値ＷからＮの減算Ｗ－Ｎが行われて第２の記憶回路に格納され、

このＷ－Ｎの減算が行われた時の演算器からボローフラグＢＲが記憶され、

ボローフラグＢＲが有るときには、第１の記憶回路と第２の記憶回路を選択する最下位アドレスを逆転させて、上記第２の記憶回路を選択するアドレスにより第１の記憶回路の演算結果Ｗを出力し、

ボローフラグＢＲが無いときには、第１の記憶回路と第２の記憶回路を選択する最下位アドレスをそのまましして、上記第２の記憶回路を選択するアドレスにより第２の記憶回路の演算結果Ｗ－Ｎを出力させることを特徴とするＩＣカード。

【請求項２５】　請求項１４において、

上記暗号化処理用演算ユニットは、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、Ｙのビットの値に応じて、Ａ＝$A^2\ R^{-1}$ｍｏｄＮ、Ａ＝ＡＢR^{-1}ｍｏｄＮの演算を行うとともに、

演算結果にオーバーフローが発生した場合にはさらに上記演算結果ＷからＮの減算Ｗ－Ｎを行なう正規動作と、各々の演算結果にオーバーフローが発生しない場合でも上記減算Ｗ－Ｎに対応した演算による無効データを生成する攪乱目的のダミー動作を行い、

上記オーバーフローの有無に対応して有効なデータを出力させることを特徴とするマイクロコンピュータ。

【請求項２６】　請求項２５において、

上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗは第１の記憶回路に格納され、

演算器からのオーバーフローフラグＯＶの有無が記憶され、

上記剰余乗算の後に上記第１記憶回路の演算結果ＷからＮの減算Ｗ－Ｎが行われて、その演算結果が上記オーバーフローフラグＯＶが有る時には上記第１の記憶回路に格納され、オーバーフローフラグＯＶが無い時には上記第１記憶回路とは異なる第２の記憶回路に上記錯乱目的のダミー動作として書き込まれ、

上記第１の記憶回路の演算結果が有効なデータとして出力されることを特徴とするマイクロコンピュータ。

【請求項２７】　請求項２５において、

上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗは第１の記憶回路に格納され、

演算器のオーバーフローフラグＯＶの有無が記憶され、

上記剰余乗算の後に上記第１の記憶回路の演算結果ＷからＮの減算Ｗ－Ｎが行われて、オーバーフローフラグＯＶが有るときに上記演算結果Ｗ－Ｎがセレクタにより選択され、オーバーフローフラグＯＶが無いときには上記第１記憶回路の演算結果Ｗがセレクタにより選択されて第２の記憶回路に格納されて有効なデータとして出力されること特徴とするマイクロコンピュータ。

【請求項２８】　請求項２５において、

上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗは第１の記憶回路に格納され、

演算器のオーバーフローフラグＯＶの有無が記憶され、

上記剰余乗算の後に上記第１の記憶回路の演算結果ＷからＮの減算Ｗ－Ｎが行われ、オーバーフローフラグＯＶが有るときには減算結果Ｗ－Ｎが第２の記憶回路に記憶され、オーバーフローフラグＯＶが無いときには減算結果Ｗ－Ｎが第３の記憶回路に記憶され、

オーバーフローフラグＯＶが有るときには上記第２の記憶回路のデータが有効なデータとして出力され、

オーバーフローフラグＯＶが無いときには上記第１の記憶回路のデータが有効なデータとして出力されること特徴とするマイクロコンピュータ。

【請求項２９】　請求項２５において、

上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗは第１の記憶回路に格納され、

演算器のオーバーフローフラグＯＶの有無が記憶され、

上記剰余乗算の後に上記第１の記憶回路の演算結果ＷからＮの減算結果Ｗ－Ｎが第２の記憶回路に格納され、オーバーフローフラグＯＶが無いとき第１の記憶回路と第２の記憶回路を選択する最下位アドレスを逆転させて、上記第２の記憶回路を選択するアドレスにより第１の記憶回路を選択して有効なデータとして出力させ、オーバーフローフラグＯＶが有るとき第１の記憶回路と第２の記憶回路を選択する最下位アドレスをそのままにして第２の記憶回路の演算結果を有効なデータとして出力させることを特徴とするマイクロコンピュータ。

【請求項３０】　請求項２５において、

上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗは第１の記憶回路に格納され、

演算器のオーバーフローフラグＯＶの有無が記憶され、

上記剰余乗算の後に上記第１の記憶回路と第２の記憶回路のアドレスが交換され、第２の記憶回路を選択するアドレスにより選択される演算結果値ＷからＮの減算Ｗ－Ｎが行われて第１の記憶回路を選択するアドレスにより選択される第２の記憶回路に減算結果Ｗ－Ｎが格納され、オーバーフローフラグＯＶが有るときにのみ上記アドレスを再度交換し、第１の記憶回路を選択するアドレスにより選択される第１又は第２の記憶回路のデータを有効なデータとして出力させること

を特徴とするマイクロコンピュータ。

【請求項３１】　請求項２５において、

上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗは第１の記憶回路に格納され、

上記剰余乗算の後に上記第１の記憶回路の演算結果値ＷからＮの減算Ｗ－Ｎが行われて第２の記憶回路に格納され、

このＷ－Ｎの減算が行われた時の演算器からボローフラグＢＲが記憶され、

ボローフラグＢＲが有るときには、第１の記憶回路と第２の記憶回路を選択する最下位アドレスを逆転させて、上記第２の記憶回路を選択するアドレスにより第１の記憶回路の演算結果Ｗを出力し、

ボローフラグＢＲが無いときには、第１の記憶回路と第２の記憶回路を選択する最下位アドレスをそのまましまして、上記第２の記憶回路を選択するアドレスにより第２の記憶回路の演算結果Ｗ－Ｎを出力させることを特徴とするマイクロコンピュータ。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

この発明は、ＩＣカードとマイクロコンピュータに関し、特にＩＣカードやプログラム内蔵の１チップマイクロコンピュータのようなＣＰＵとメモリを含み暗号鍵を使ったデータ処理を行なうものの機密保護技術に利用して有効な技術に関するものである。

【０００２】

【従来の技術】

メモリに保存されている鍵情報を用いてデータの暗号処理化又は復号化処理を行なうようにしたＩＣカードにおいて、処理時間の違いを利用して実行内容や暗号鍵を推定するＴＡ（Timing Attack ）法のようなハッキング手法に対抗するため、暗号処理化又は復号化処理の実行中又は実行の前後に、鍵情報の内容との時間的な相関関係を喪失させる遅延処理を実行する技術の例として、特開平１０－６９２２２号がある。また、ＩＣカードに関しては、オーム社出版電子情報通信

学会編水沢順一著「ＩＣカード」などがある。

【０００３】

【発明が解決しようとする課題】

近年、ＩＣカードが暗号処理を行っている時の消費電流を観測して解析することにより、容易に暗号処理の内容や暗号鍵が推定されることの可能性が示唆されている。このことについては、John Wiley & sons 社 W.Rankl & W. Effing著「Smart Card Handbook 」の8.5.1.1 Passive protective mechanisms(263ページ) に記載されている。

【０００４】

つまり、ＳＰＡ (Simple Power Analysis) 法では、演算命令の違い、あるいは処理されているデータの違いにより生じる消費電流波形の違いから、暗号鍵や処理されているデータを解析し、ＤＰＡ (Differential Power Analysis) 法では、消費電流波形を統計処理して暗号鍵を推定する。このＤＰＡ法では、例えばＤＥＳのある部分に仮定した暗号鍵をあてはめて、平文を変化させながら消費電流波形を測定して統計する。暗号鍵を様々に変化させながらこの作業を繰り返し、正しい鍵のときには電流波形が大きなピークを示す。

【０００５】

前記公報に記載のようにＴＡ (Timing Attack) 法のみを考慮した遅延処理では、実際の演算による消費電流の相関性までも喪失させることができず、上記のような消費電流波形を観測するというＳＰＡ又はＤＰＡ法のようなハッキング手法には対抗できない。そこで、本願発明者等においては、上記ＩＣカード及びＩＣカード等のようなモジュールに搭載されるマイクロコンピュータのように内蔵のプログラムにより一定のデータ処理動作を行うものに対して上記のような消費電流の観測による暗号処理の内容や暗号鍵の解読をより確実に防止することができる機密保護技術を開発するに至った。

【０００６】

この発明の目的は、機密保護の強化を実現したＩＣカードとマイクロコンピュータを提供することにある。この発明の他の目的は、機密保護のための信号処理の高速化とその強化を実現したＩＣカードとマイクロコンピュータを提供するこ

とにある。この発明の前記ならびにそのほかの目的と新規な特徴は、本明細書の記述および添付図面から明らかになるであろう。

【０００７】

【課題を解決するための手段】

本願において開示される発明のうち代表的なものの概要を簡単に説明すれば、下記の通りである。すなわち、外部端子がリードライト装置と電気的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ったデータの入出力動作を含むＩＣカードにおいて、上記暗号化処理又は復号化処理に本来の処理動作に似た攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにする。

【０００８】

本願において開示される発明のうち他の代表的なものの概要を簡単に説明すれば、下記の通りである。すなわち、暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータにおいて、上記暗号化処理又は復号化処理に本来の処理動作に似た攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにする。

【０００９】

【発明の実施の形態】

図１には、この発明が適用されるＩＣカードの一実施例の外観図が示されている。ＩＣカードは、プラスチックケースからなるカード１０１と、かかるカード１０１の内部に搭載された図示しない１チップのマイクロコンピュータ等からなるＩＣカード用チップを持つものである。上記ＩＣカードは、さらに上記ＩＣカード用チップの外部端子に接続されている複数の接点（電極）１０２を持つ。複数の接点１０２は、後で図２によって説明するような電源端子ＶＣＣ、電源基準電位端子ＶＳＳ、リセット入力端子ＲＥＳバー、クロック端子ＣＬＫ、データ端子Ｉ／Ｏ－１／ＩＲＱバー、Ｉ／Ｏ－２／ＩＲＱバーとされる。ＩＣカードは、かかる接点１０２を通して図示しないリーダーライタのような外部結合装置から電源供給を受け、また外部結合装置との間でのデータの通信を行う。

【００１０】

図２には、この発明に係るＩＣカードに搭載されるＩＣカード用チップ（マイクロコンピュータ）の一実施例の概略ブロック図が示されている。同図の各回路ブロックは、公知のＭＯＳ集積回路の製造技術により、特に制限されないが、単結晶シリコンのような１個の半導体基板上において形成される。

【００１１】

この発明に係るＩＣカード用チップの構成は、基本的にマイクロコンピュータと同じような構成である。その構成は、クロック生成回路２０５、中央処理装置（以下単にＣＰＵという場合がある）２０１、ＲＯＭ(Read Only Memory)２０６やＲＡＭ(Random Access Memory)２０７、不揮発性メモリ２０８などの記憶装置、暗号化及び復号化処理の演算を行なうコプロセッサ２０９、入出力ポート（Ｉ／Ｏポート）２０２などからなる。

【００１２】

クロック生成回路２０５は、図示しないリーダライタ（外部結合装置）から図１の接点１０２を介して供給される外部クロックＣＬＫを受け、かかる外部クロック信号に同期したシステムクロック信号を形成し、それをチップ内部に供給する回路である。ＣＰＵ２０１は、論理演算や算術演算などを行う装置であり、システムコントロールロジック、乱数発生器及びセキュリィロジック及びタイマなどを制御する。記憶装置２０６、２０７、２０８は、プログラムやデータを格納する装置である。コプロセッサ２０９は、後述するようにＲＳＡ暗号法などに応用可能なべき乗剰余乗算動作を行なう演算器とレジスタ及び制御論理から構成される。Ｉ／Ｏ（入出力）ポート２０２は、リーダライタと通信を行う装置である。データバス２０４とアドレスバス２０３は、各装置を相互に接続するバスである。

【００１３】

上記記憶装置２０６，２０７，２０８のうち、ＲＯＭ２０６は、記憶内容が不揮発的に固定されているメモリであり、主にプログラムを格納するメモリである。揮発性メモリ（以下、ＲＡＭという）２０７は自由に記憶情報の書き換えができるメモリであるが、電源の供給が中断されると、記憶している内容が消えてなくなる。ＩＣカードがリーダライタから抜かれると電源の供給が中断されるため

、ＲＡＭ２０７の内容は、保持されなくなる。

【００１４】

上記不揮発性メモリ（以下、ＥＥＰＲＯＭ(Electrical Erasable Programmable Read Only Memory)という）２０８は、内容の書き換えが可能な不揮発性メモリであり、その中に一旦書き込まれた情報は、電源の供給が停止されてもその内部に保持される。このＥＥＰＲＯＭ２０８は、書き換える必要があり、かつＩＣカードがリーダライタから抜かれても保持すべきデータを格納するために使われる。例えば、ＩＣカードがプリペイドカードとして使用されるような場合、のプリペイドの度数などは、使用するたびに書き換えられる。この場合の度数などは、リーダライタか抜かれてもＩＣカード内で記憶保持する必要があるため、ＥＥＰＲＯＭ２０８で保持される。

【００１５】

ＣＰＵ２０１は、いわゆるマイクロプロセッサと同様な構成にされる。すなわち、その詳細を図示しないけれども、その内部に命令レジスタ、命令レジスタに書込まれた命令をデコードし、各種のマイクロ命令ないしは制御信号を形成するマイクロ命令ＲＯＭ、演算回路、汎用レジスタ（ＲＧ６等）、内部バスＢＵＳに結合するバスドライバ、バスレシーバなどの入出力回路を持つ。ＣＰＵ２０１は、ＲＯＭ２０６などに格納されている命令を読み出し、その命令に対応する動作を行う。ＣＰＵ２０１は、Ｉ／Ｏポート２０２を介して入力される外部データの取り込み、ＲＯＭ２０６からの命令や命令実行のために必要となる固定データのようなデータの読み出し、ＲＡＭ２０７やＥＥＰＲＯＭ２０８に対するデータの書き込みと読み出し動作制御等を行う。

【００１６】

上記ＣＰＵ２０１は、クロック生成回路２０５から発生されるシステムクロック信号を受けそのシステムクロック信号によって決められる動作タイミング、周期をもって動作される。ＣＰＵ２０１は、その内部の主要部がＰチャンネル型ＭＯＳＦＥＴとＮチャンネル型ＭＯＳＦＥＴとからなるＣＭＯＳ回路から構成される。特に制限されないが、ＣＰＵ２０１は、ＣＭＯＳスタティックフリップフロップのようなスタティック動作可能なＣＭＯＳスタテック回路と、信号出力ノー

ドへの電荷のプリチャージと信号出力ノードへの信号出力とをシステムクロック信号に同期して行うようなＣＭＯＳダイナミック回路とを含む。

【００１７】

ＩＣカードのセキュリティ機能としては、チップ内部で乱数を自動生成する乱数発生器や、ランダムに割込みを生成するタイマー機能などの他に、本願発明にかかる高セキュリテイ機能として、ＩＣカードと外部装置とのデータ送受信の際に用いるＲＳＡ暗号法などに応用可能なべき乗剰余演算動作を行なう暗号処理用演算ユニット（コプロセッサ）２０９を内蔵している。このコプロセッサ２０９は専用のレジスタが内蔵されている。

【００１８】

ＩＣカードにおけるセキュリティ・システムでは、通信データの暗号処理は必須であり、この実施例でも現在最も多く利用されている公開鍵暗号としてＲＳＡ暗号が用いられる。この暗号法では、暗号化・復号化ともにべき乗剰余乗算X^Y ｍｏｄＮを用いるが、これは古くから知られている計算アルゴリズムによって剰余乗算A^2 ｍｏｄＮとＡＢｍｏｄＮの２つの形に分解することができる。つまり、Ｙ＝e_n e_{n-1} ・・・e_1 の値e_i を上位e_n から最下位のe_1 まで順に１ビットずつ見ていき、e_i ＝０だったらA^2 ｍｏｄＮのみを、e_i ＝１だったらA^2 ｍｏｄＮとＡＢｍｏｄＮを演算する。したがって、e_i ＝０のときにはA^2 ｍｏｄＮの演算の後にｉ＝０であるかの判定処理が行なわれ、e_i ＝１のときにはA^2 ｍｏｄＮとＡＢｍｏｄＮとの演算の後にｉ＝０であるかの判定処理が行なわれるために、e_i ＝０と１とに対応した２通りの電流波形の形態が現れてしまう。

【００１９】

この実施例のようにコプロセツサ２０９を用いた場合には、その消費電流はＣＰＵの消費電流に比べて比較的大きいため、この部分の電流波形を観測することによりコプロセッサの動作形態を比較的容易に識別することができ、前記ＴＡ法とＳＰＡ法により暗号鍵Ｙの値を解読されてしまう可能性が高い。そこで、この実施例のコプロセッサ２０９では、上記暗号化・復号化ともに用いられるべき乗剰余乗算X^Y ｍｏｄＮの演算を行なうに当たり攪乱目的のダミーの演算が挿入さ

れる。つまり、図３のタイミング図及び図４のフローチャート図に示すようにe_i＝０でも１でもA^2 ｍｏｄＮとＡＢｍｏｄＮの両方の演算を常に行なうようにするものである。

【００２０】

図３のタイミング図において、（ａ）に示すように本来は、e_n＝１のときにはA^2 ｍｏｄＮの演算を行い、e_nの判定の１により時間ｔ１を経てＡＢｍｏｄＮの演算を行い、その演算後にｉをデクリメント（ｎ－１）してｉ＝０の判定に時間ｔ２を費やす。次いで、次ビットe_{n-1}＝０のときは、A^2 ｍｏｄＮの演算を行い、e_{n-1}＝０の判定とｉをデクリメント（ｎ－２）してｉ＝０の判定に時間ｔ３を費やす。そして、次ビットe_{n-2}＝１のときには、A^2 ｍｏｄＮの演算を行い、e_{n-2}の判定の１により時間ｔ１を経てＡＢｍｏｄＮの演算を行い、その演算後にｉをデクリメント（ｎ－３）してｉ＝０の判定に時間ｔ２を費やす。以下、同様にe_1まで同様な動作を繰り返すものである。

【００２１】

この実施例のコプロサッサ２０９においては、上記暗号鍵Ｙの各ビットe_iの論理０又は１に無関係にA^2 ｍｏｄＮの演算の後にＡＢｍｏｄＮの演算を行なうようにする。図３（ｂ）のe_{n-1}＝０のときのようにe_iが論理０のときにおける上記ＡＢｍｏｄＮの演算が攪乱目的のダミー演算として挿入される。つまり、（ｂ）のタイミング図及び図４のフローチャート図のように、A^2 ｍｏｄＮとＡＢｍｏｄＮの演算動作の間には、例えばe_iの判定の判定を含む時間ｔ１が費やされ、ＡＢｍｏｄＮと次ビットに対応したA^2 ｍｏｄＮの演算動作の間には、ｉのデクリメント動作とｉ＝０の判定時間ｔ２が費やされる画一化された動作タイミング及び動作電流とすることができる。ただし、この実施例では、e_iの判定処理は、その結果が演算動作の分岐の条件とされないため図４のフローチャート図では省略されている。

【００２２】

図５には、上記コプロセッサの一実施例のブロック図が示されいてる。この実施例では、主に演算器、制御論理、専用レジスタブロックより構成され、べき乗剰余演算の最終結果はデータバツファ、データバスを介して中央処理装置ＣＰＵ

に送信される。専用レジスタは、アドレスバスから供給されるアドレス信号に対応してその選択動作が行なわれる。

【００２３】

この実施例では、内部バスＭＤＢとレジスタブロックのリードライトバッファ（Ｒ／Ｗ　Ｂuffer)との間にゲート回路１が設けられる。このゲート回路１は、制御論理により制御が行なわれて、e_i が論理０ならばA^2 ｍｏｄＮ動作の演算結果が内部バスＭＤＢとリードライトバッファを介して所定のレジスタＣＤＡに取り込まれた後開いていたゲートが閉じるようにされる。つまり、上記演算結果がリードライトバッファに取り込まれると、その後にゲートを閉じてしまいリードライトバッファへの新たなデータの書き込みを禁止する。したがって、その後に行なわれるＡＢｍｏｄＮの演算結果は無効データとして扱われることとなる。また、e_i が論理１ならばゲート回路１はゲートを開いた状態のままとされる。

【００２４】

図６には、上記コプロセッサの他の一実施例のブロック図が示されている。この実施例では、レジスタブロックのリードライトバッファ（Ｒ／Ｗ　Ｂuffer)と各レジスタとの間にゲート回路２が設けられる。このゲート回路２は、前記同様に制御論理により制御が行なわれて、e_i が論理０ならばA^2 ｍｏｄＮ動作の演算結果が内部バスＭＤＢとリードライトバッファとを介して所定のレジスタＣＤＡに書き込まれた後に開いていたゲートが閉じるようにされる。つまり、上記演算結果がレジスタＣＤＡに取り込まれると、その後にゲートを閉じてしまいかかかるレジスタＣＤＡへの新たなデータの書き込みを禁止する。したがって、その後に行なわれるＡＢｍｏｄＮの演算結果は、リードライトバッファまでは書き込まれるが、実際には無効データとして扱われることとなる。また、e_i が論理１ならばゲート回路２はゲートを開いた状態のままとされる。

【００２５】

図７には、上記ゲート回路の一実施例の内部構成図が示されている。ダミー書き込み制御ユニットは、アンドゲート回路によって構成され、一方の入力には制御論理からのライトイネーブル信号が供給され、他方の入力には演算器で生成されたライトストローブ信号が供給される。上記ゲート回路の出力信号は、データ

バッファ（Ｒ／Ｗ　Ｂuffer)と専用レジスタにライトスローブ信号として伝えられる。

【００２６】

この実施例では、演算結果そのものの伝達制御するものに代えて、レジスタ又はデータバッファへの書き込み動作を指示するライトストローブ信号の発生タイミングを切り換えるようにするものである。つまり、e_i＝０のときには、A^2 ｍｏｄＮ動作の演算結果が出力された後にライトイネーブル信号をロウレベルとしてアンドゲート回路のゲートが閉じるようにするものである。逆に、e_i＝１のときには、制御論理はライトイネーブル信号をハイレベルのままとして、演算器で形成されたライトストローブ信号がそのままデータバッファ又は専用レジスタに伝えられる。この構成では、複数ビットからなる演算結果Ａに対応して、複数個のゲート回路を設ける必要がないので簡素化が可能になる。

【００２７】

図８には、上記コプロセッサの他の一実施例のブロック図が示されいてる。この実施例では、レジスタブロックのリードライトバッファ（Ｒ／Ｗ　Ｂuffer)と各レジスタとの間にセレクタ２とレジスタブロックにダミーレジスタ１が設けられる。このセレクタ２は、前記同様に制御論理により制御が行なわれて、e_iが論理０ならばA^2 ｍｏｄＮ動作の演算結果が内部バスＭＤＢとリードライトバッファとを介して所定のレジスタＣＤＡに書き込まれるような信号経路を形成し、その後にダミーレジスタ１を選択するような信号経路を形成する。

【００２８】

つまり、上記演算結果がレジスタＣＤＡに取り込まれると、その後にダミーレジスタ１を選択するので、レジスタＣＤＡへの新たなデータの書き込みを禁止しつつその後に行なわれるＡＢｍｏｄＮの演算結果がダミーレジスタに書き込まれものとなる。e_iが論理１ならばセレクタ２は常にレジスタＣＤＡを選択する。この構成は、演算結果をレジスタに書き込む動作を含めてe_iが論理０のときと論理１のときとで電流波形でみたときに全く同一にすることができるから、電流波形を利用したアタックをより確実に無力化することができる。

【００２９】

図９には、この発明に係るコプロセッサの他の一実施例の動作を説明するための構成図が示されている。図９（ａ）のタイミング図及び（ｂ）のフローチャート図において、前記説明したように、A^2 ｍｏｄＮの演算後、e_i の判定の時間ｔ１の間もダミー演算動作としてA^2 ｍｏｄＮを継続してＡＢｍｏｄＮの演算に移行する。

【００３０】

その演算後にｉをデクリメント（－１）してｉ＝０の判定に時間ｔ２を費やすが、その間も上記ＡＢｍｏｄＮの演算を継続させる。以下、同様にe_1 まで同様な動作を繰り返すものである。この構成は、演算動作中は、e_i が論理０と１のときに関係なく上記のような演算動作を継続するので、電流波形でみたときに格別な特徴を見出すことができないから、電流波形を利用したアタックを無力化することができる。

【００３１】

図１０には、図９のコプロセッサの動作を実現するための一実施例のブロック図が示されている。制御論理では、ダミーイネーブル信号とコプロイネーブル信号を送出する。上記ダミーイネーブル信号とコプロイネーブル信号は、オアゲート回路を通して演算器に入力される。それ故、コプロイネーブル信号がアクティブであるときに加えて、ダミーイネーブル信号がアクティブであるときにも演算器は演算動作を行なうようにされる。

【００３２】

上記ダミーイネーブル信号は、インバータ回路を通してアンドゲート回路の一方の入力に供給され、かかるアンドゲート回路の他方の入力には演算器で形成されたライトストローブ信号が供給される。つまり、演算器で形成されたライトストローブ信号の伝達をダミーイネーブル信号で選択的に停止できるようにする。コプロイネーブル信号がアクティブにされて、前記正規の演算動作が終了すると、その演算結果を出力するためのライトストローブ信号が形成される。このようにコプロイネーブル信号がアクティブのときには、ダミーイネーブル信号の反転信号がアクティブレベルとなってアンドゲート回路のゲートを開くように制御するので、上記正規演算結果はライトストローブ信号によって、Ｒ／Ｗバッファ又

はレジスタブロックの所定のレジスタに書き込まれる。

【００３３】

上記のような正規演算が終了すると、ダミーイネーブル信号がアクティブとなって演算器に対して演算動作を指示する。この演算の終了によって、上記ライトストローブ信号が形成されるが、上記ダミーイネーブル信号の反転信号によってアンドゲート回路がゲートを閉じているので、上記攪乱目的のダミー演算動作によって発生されたライトストローブ信号がＲ／Ｗバッファ又はレジスタブロックの所定のレジスタに伝えられることはない。これにより、攪乱目的のダミー演算結果は無効データとして消失させられる。

【００３４】

図１１には、この発明に係るコプロセッサの他の一実施例の動作を説明するためのタイミング図が示されている。前記図３に示した実施例のように、攪乱目的のダミー演算を挿入して、（ａ）のタイミング図のように、e_i に対して画一化してA^2 ｍｏｄＮとＡＢｍｏｄＮの演算を一対として行なうようにした場合でも、各演算には、演算結果にオーバーフロー処理を必要とするもの（あり）のものと、オーバーフロー処理を必要としないもの（なし）が発生する。

【００３５】

このようなオーバーフロー処理は、演算時間を長くするものであるので電流波形でみると、オーバーフロー処理ありとなしとの識別が可能になる。このような電流波形の特徴から演算内容や演算データを推測することも不可能ではないと考えられるため、この実施例では（ｂ）のタイミング図に示すようにオーバーフロー処理を不要とする演算に対しても必要なときと同様にオーバーフロー処理を挿入する。つまり、みかけ上は、全ての演算A^2 ｍｏｄＮとＡＢｍｏｄＮの演算において画一的にオーバーフロー処理のための動作が実施されるために、その識別を無力化するものである。

【００３６】

図１２は、この発明に係るコプロセッサの他の一実施例の動作を説明するためのフローチャート図が示されている。このフローチャート図は、前記図１１（ｂ）に対応している。A^2 ｍｏｄＮとＡＢｍｏｄＮの各演算は、剰余演算部とオー

バーフロー演算部からなり、演算結果に無関係に上記オーバーフロー演算処理を実施するものである。

【００３７】

図１３には、この発明に係るコプロセッサの他の一実施例の動作の詳細を説明するためのタイミング図が示されている。この実施例による対策前では、前記A^2 ｍｏｄＮとＡＢｍｏｄＮのようなコプロ演算においては、その演算結果に対応してオーバーフロー処理のあるものと無いもの２種類が存在したが、この実施例による対策後では、前記A^2 ｍｏｄＮとＡＢｍｏｄＮのようなコプロ演算においては、その演算結果に無関係に常にオーバーフロー処理が実行される。このため、本来はオーバーフロー処理が不要な演算動作に対して実施されたオーバーフロー処理は、攪乱目的のダミー動作とされる。

【００３８】

図１４には、図１１ないし図１３に示したコプロセッサの動作を実現するための一実施例のブロック図が示されている。制御論理では、ダミーオーバーフロー信号とコプロオーバーフロー信号を送出する。上記ダミーオーバーフロー信号とコプロオーバーフロー信号は、オアゲート回路を通して演算器に入力される。それ故、コプロオーバーフロー信号がアクティブであるときに加えて、ダミーオーバーフロー信号がアクティブであるときにも演算器はオーバーフロー処理動作を行なうようにされる。

【００３９】

上記コプロオーバーフロー信号は、アンドゲート回路の一方の入力に供給され、かかるアンドゲート回路の他方の入力に演算器で形成されたライトストローブ信号が供給される。つまり、演算器で形成されたライトストローブ信号の伝達をコプロオーバーフロー信号がアクティブレベルでないときに選択的に停止できるようにする。つまり、コプロオーバーフロー信号がアクティブレベルでないときはダミーオーバーフロー信号によって演算器がオーバーフロー処理を行なっているので、かかるオーバーフロー処理で形成されたライトストローブ信号は上記ゲート回路のゲートを閉じることによって無効にするものである。したがって、前記正規のオーバーフロー処理終了すると、その処理結果を出力するためのライト

ストローブ信号が形成されて、Ｒ／Ｗバッファ又はレジスタブロックの所定のレジスタに処理結果が書き込まれる。

【００４０】

これに対して、ダミーオーバーフロー信号がアクティブとなって演算器に対してオーバーフロー処理動作を指示した場合には、そのオーバーフロー処理によって形成されたライトストローブ信号は、上記コプロオーバーフロー信号によってアンドゲート回路のゲートが閉じられるものであるから、上記攪乱目的のダミーオーバーフロー処理動作によって発生されたライトストローブ信号がＲ／Ｗバッファ又はレジスタブロックの所定のレジスタに伝えられることはない。これにより、攪乱目的のダミーオーバーフロー処理結果は無効データとして消失させられる。

【００４１】

図１５には、この発明に係るコプロセッサの更に他の一実施例の動作を説明するためのタイミング図が示されている。（ａ）に示すように本来は、e_n＝１のときにはA^2 ｍｏｄＮの演算を行い、e_nの判定の１により時間ｔ１を経てＡＢｍｏｄＮの演算を行い、その演算後にｉをデクリメント（ｎ－１）してｉ＝０の判定に時間ｔ２を費やす。次いで、次ビットe_{n-1}＝０のときは、A^2 ｍｏｄＮの演算を行い、e_{n-1}＝０の判定とｉをデクリメント（ｎ－２）してｉ＝０の判定に時間ｔ３を費やすような演算動作に対して、上記各演算毎の時間ｔ１、ｔ２及びｔ３に対して攪乱目的のダミーサイクルが挿入される。

【００４２】

（ｂ）のタイミング図では、上記攪乱目的のダミーサイクルの挿入は、各演算毎の時間を最も長い時間ｔ３に揃えるように挿入するものである。これにより、時間ｔ３をインターバルとしてA^2 ｍｏｄＮ又はＡＢｍｏｄＮのいずれかの演算が実施されるために、みかけ上は演算動作に対応した電流波形が画一化されてその識別を無力化するものである。これに対して、（ｃ）タイミング図では、上記（ｂ）とは逆に上記演算毎のインターバルにおいて時間がランダムに変化する攪乱目的のダミーサイクルが挿入される。上記A^2 ｍｏｄＮ又はＡＢｍｏｄＮのいずれかの演算が時間的にランダムに実施される。そのため、電流波形でみると上

記各演算動作と無関係で、かつ不規則性の電流値にされる。言い換えるならば、上記演算器において同じ状態及び同じ動作でも毎回異なるよう、統計的な観点での非再現性を持つようにされるために、その識別を無力化することができる。

【００４３】

上記のような攪乱目的のダミーサイクルは、前記図２に示されたようにタイマーを利用して演算間隔を可変にするものである。あるいはコプロセッサの外部にタイマーを設けて一定の時間が経過するまで次の演算の実行を待つようにする。つまり、コプロセッサによるべき乗剰余乗算の演算において、図１５（ａ）に示した前記各演算毎の時間ｔ１，ｔ２，ｔ３に攪乱目的のダミーのサイクルを挿入し、一定時間後にタイマーからの割込みを入れる。これにより、図１５（ｂ）に示すようにｔ１，ｔ２，ｔ３の時間が全て一定となり、電流波形からのアタックを困難にする。あるいはタイマーには乱数発生器で生成した乱数をセットしておき、（ｃ）に示すように毎回ｔ１，ｔ２，ｔ３の時間をランダムに変化させることも可能である。また、タイマーを用いなくても、ソフトウエアでカウントすることも可能である。

【００４４】

べき乗剰余乗算において、コプロセッサによる演算の高速化を目的とし、Ｙの値を２ビット、あるいは３ビットずつ処理するようにすると、例えば図１６のフローチャート図に示すように、２ビット処理の例で説明するなら常にA^2 ｍｏｄＮ－A^2 ｍｏｄＮ－ＡＢｍｏｄＮ及びｉ－２とｉ＝０？の各ステップの繰り返しになるので、前記１ビットずつ行なう場合のような攪乱目的のダミー演算を行なわなくとも、処理時間や電流波形が一定になる。そのため、電流波形からＹの値を推定するのは困難になる。また演算の回数も、前記のバイナリ法だと最大で２ｎ回かかっていたものを、２ビット処理だと常に１．５ｎ回で済むために、動作時間の短縮にもつながる。

【００４５】

コプロセッサの演算が開始するまでにＡ，Ｂ，Ｎの値をそれぞれコプロセツサ専用レジスタに転送し格納しておく。しかしながら、２ビット処理を行う場合、、Ｙの値によって４通りのＢの値B_1、B_2、B_3、B_4が必要になり、これら

の値は前もって計算して．ＲＡＭやＥＥＰＲＯＭなどに格納しておき、毎回コプロセッサ専用レジスタに転送することになるが。この際、４通りのＢの値によって転送中の電流波形に特徴が現れる可能性がある。

【００４６】

例えば、１６ビットのプリチャージバスにデータを転送する場合を考える。プリチャージバスは、データ転送の前にすべてのバスの値を“１”にそろえるバスである。このバスに、値は違うが“１”のビットの数が同じデータ、例えば、“１”のビットの数が２である１６進数で“８８”と“１１”、を転送した場合、電流波形はほぼ同じ波形になると予測される。この理由は、“１”から“０”へ変化したビットの数が同じであるため、同じように電流を消費し、同じ電流波形になるからである。

【００４７】

もし、“１”のビットの数が１つ異なるデータ、例えば、“１”のビットの数が３である“８９”や“１９”を転送した場合、“１”のビットの数が２のデータとは消費電流が異なる。これは、１３ビット分バスの値が“１”から“０”に変わったため、その分の電流が消費される。そのため、先の１４ビットが変化したデータに比べて消費電流が１ビット分小さくなる。一般に、変化するビットの数が多いほど電流波形は高くなるという規則性がある。この規則性から転送されているデータを推定することができると思われる、電流アタックの対象となりやすい。これを防ぐため次のような工夫を行なうものである。

【００４８】

図１７と図１８には、この発明に係るコプロセッサの他の一実施例のブロック図がそれぞれ示されている。この実施例のコプロセッサは、２ビット処理と３ビット処理に向けられている。つまり、コプロセッサのレジスタ容量を増やして、２ビット処理の場合には４通りのＢの値B_1～B_4を、３ビット処理の場合には８通りのＢの値B_1～B_8をコプロセッサのレジスタに格納しておく。従って、演算の途中で記憶回路（ＲＡＭ）からデータバスを通して上記コプロセッサのレジスタに前記のような転送の必要がなくなり、前記電流アタックに対して防御することができる。

【００４９】

制御用レジスタ（ＣＣＮＴ）

ビット７	ビット６		ビット２	ビット１	ビット０
－	－	……		e_i	e_{i-1}

【００５０】

演算の種類

ビット２	e_i	e_{i-1}	演算の種類
０	０	０	$A \leftarrow A^2 \bmod N$
０	１	０	$A \leftarrow A \bmod N$
０	１	１	$A \leftarrow A \times N$
１	０	０	$A \leftarrow AB_1 \bmod N$
１	０	１	$A \leftarrow AB_2 \bmod N$
１	１	０	$A \leftarrow AB_3 \bmod N$
１	１	１	$A \leftarrow AB_4 \bmod N$

【００５１】

つまり、前記図１６に示したようなフローチャート図において、コプロセッサがＡＢｍｏｄＮを実行する際、下記のように４つ（３ビット処理のときにはあるいは８つ）のうちの正しいＢレジスタＣＤＢから値を選んで実行できるように、Ｙの２ビット（あるいは３ビット）の値をコプロセッサの制御レジスタ（ＣＣＮＴ）のビットに当てはめ、前記に示す制御レジスタ及び演算の種類のように、２ビット処理の場合には、AB_1 ｍｏｄＮ，AB_2 ｍｏｄＮ，AB_3 ｍｏｄＮ，AB_4 ｍｏｄＮのうちどの演算をするかを選択させるようにする。

【００５２】

図１９には、この発明に係るコプロセッサの他の一実施例のブロック図が示さ

れている。この実施例のコプロセッサも、２ビット処理や３ビット処理のような複数ビット処理に向けられている。この実施例では、データバスにスイッチを設けて演算をしながら転送できるようにする。この構成により、コプロセッサのレジスタ容量を増加させることなく、実行時間の短縮と電流アタック対策の両方に効果的である。

【００５３】

コプロセツサ専用レジスタ（ＣＤＡ，ＣＤＢ，ＣＤＮ，ＣＤＷ）は、同図に示すように４つのレジスタがＣＰＵとコプロセッサの演算器との間で排他的に使用されている。２ビット処理を行う場合、２回のA^2 ｍｏｄＮを行いながらその間にＢの値をＲＡＭからコプロセッサ専用レジスタユニット中のＢレジスタＣＤＢに転送できるようにすると効率的である。

【００５４】

コプロセッサのＡレジスタＣＤＡとＢレジスタＣＤＢのＩ／Ｏを分け、それぞれにリード／ライトバッファ（Ｒ／Ｗ　Ｂuffer)を設けて、それぞれ独立に動作できるようにする。演算器がA^2 ｍｏｄＮを演算している間は、制御信号によりデータバスをパス１（ｐａｔｈ１）につなぎ、図示しないＣＰＵのＲＡＭからＢの値を上記独立に設けられたリード／ライトバッファを介してＢレジスタＣＤＢに転送する。次に演算器がＡＢｍｏｄＮを実行する際には、制御信号によりパス２（ｐａｔｈ２）に切り換え、上記Ｂレジスタの Ｂ値を演算器に送り上記ＣＰＵがＢレジスタＣＤＢにアクセスできないようにする。この方法を取ると、A^2 ｍｏｄＮを演算動作と、Ｂ値の転送動作が同時に行なわれるから演算時間が短縮されるだけでなく、演算と転送の消費電流が重なるため双方の波形が識別できなくなり、電流アタック対策に有効である。

【００５５】

図２０には、この発明に係るＩＣカード用チップの他の一実施例の要部ブロック図が示されている。この実施例では、暗号処理用演算ユニットとメモリ（ＲＡＭ）間の転送の際、メモリにカウンタを設けるようにするものである。この実施例では、２ビット処理に用いる４通りの値、あるいは３ビット処理に用いる８通りの値をコプロセッサ外部メモリＲＡＭからコプロセッサ専用レジスタユニット

中のＢレジスタＣＤＢに転送する際の電流撹乱を行なうようにするものである。

【００５６】

この実施例では、前記図２に示したようなＩＣカード用チップにおいて、ＲＡＭの側にカウンタが設けられる。ＲＡＭは、カウンタで形成されたアドレス信号をデコードしてデータをデータバスに送出する。このとき、アドレスバスには、乱数発生器が形成された偽アドレスが送出される。これにより、アドレスとデータとの相関が無くなり、電流解析を困難とさせることができる。

【００５７】

図２１には、上記カウンタの一実施例のブロック図が示されている。カウンタは、転送したいブロックの最初のアドレスを保持する先頭アドレスレジスタとインクリメンタを用い、ブロック転送をイネーブルにするイネーブル信号とクロック又はリード／ライト信号などによるインクリメント指示信号で制御する。ブロック転送を開始する際、まず転送の先頭アドレスと転送開始のイネーブル信号がＣＰＵよりカウンタに送信され、上記先頭アドレスレジスタに保持される。その後は、インクリメント指示信号によって、インクリメンタが動作して先頭アドレスレジスタの先頭アドレスＡ＋１を形成して、アドレスを生成するとともに上記先頭アドレスレジスタの内容を書き換えるので、図２２のタイミング図に示すように、ＲＡＭアドレスが順番にインクリメントＡ、Ａ＋１、Ａ＋２、・・・されていき、そのアドレスに従って順次データD_A、D_{A+1}、D_{A+2}・・・・が書込まれ／読み出される。

【００５８】

この実施例では、ブロック転送がイネーブルになった後はアドレスバスからのアドレスをカウンタが受け付けないため、アドレスバスにどのような値が来ようとデータは正しく読み出されていく。従って、アドレスバスに乱数発生器などで生成した乱数Ｂ、Ｃ、Ｄ、Ｅ・・・が出力されるとアドレスバスの消費電流を撹乱でき、この効果からチップ全体の消費電流を撹乱できるため、チップ内部動作の解析を困難にすることが可能になる。

【００５９】

図２３には、この発明に係るＩＣカード用チップの更に他の一実施例を示す要

部ブロック図が示されている。この実施例でも、暗号処理用演算ユニットとメモリ（ＲＡＭ）間の転送の際、メモリにカウンタを設けるようにするものであが、かかる暗号処理用演算ユニットとメモリＲＡＭの最初のアドレスをも撹乱するようアドレスオフセット機能が設けられる。つまり、乱数発生器などで生成した乱数をあらかじめＣＰＵとカウンタ側に同時に転送しておき、ブロック転送の最初のアドレスに乱数を加えるか又は引くかした値をアドレスバスに出力する。カウンタ側ではアドレスバスの値を同じ乱数を用いて復号化し、最初のアドレスを得る。

【００６０】

図２４には、上記転送動作を説明するためのタイミング図が示されている。乱数発生器で形成された乱数をあらかじめＣＰＵとＲＡＭに転送しておき、オフセット演算部１によりブロック転送の最初のアドレスＡに乱数Ｓを加えるか引くかしたアドレスＡ±Ｓをアドレスバスに送出する。カウンタ側では、アドレスバスの値を同じ乱数Ｓを用いて復号化し、オフセット演算部２により最初のアドレスＡを得て、以後前記同様にインクリメントしてアドレスＡ＋１、Ａ＋２・・・を生成する。このようなアドレスＡ＋１、Ａ＋２に同期して、乱数発生器が乱数Ｂ、Ｃ、Ｄ・・・をアドレスバスに送出するので、先頭のアドレスを含めてアドレスバスの消費電流を撹乱でき、チップ内部動作の解析をいっそう困難にすることが可能になる。

【００６１】

前記実施例のような暗号化／復号化装置において、べき乗剰余演算「X^Y ｍｏｄＮ」（Ｘ，Ｙ，Ｎは正の整数）を用いた場合、Ｘ，Ｙ，Ｎが、通常１００ビット～２０００ビット程度の非常に大きな数が使用されるため、「X^Y ｍｏｄＮ」をいかにして高速に実行するかが重要となる。その一つの解法として、剰余乗算「ABR^{-1}ｍｏｄＮ」を実行する次のようなアルゴリズムが知られており、本願出願人においては、特開平１０－２１０５７号公報（米国登録番号５，９６１，５７８）において、「ABR^{-1}ｍｏｄＮ」のアルゴリズムを基にした積和演算器を用いたマイクロコンピュータを提案している。

【００６２】

上記アルゴリズムは、次のステップ(1) ないし(12)からなる。

(1)　ｉｎｐｕｔ　Ｘ，Ｙ＝ e_n e_{n-1} ･･･ e_1 ，Ｎ，Ｒ

(2)　Ｂ＝R^2 ｍｏｄＮ

(3)　Ａ＝Ｘ

(4)　Ａ＝ＡＢR^{-1}ｍｏｄＮ＋ｋＮ

(5)　Ｂ＝Ａ

(6)　ｆｏｒ　ｉ＝ｎ－１　ｔｏ　１　step－１｛

(7)　Ａ＝A^2 R^{-1}ｍｏｄＮ＋ｋＮ

(8)　ｉｆ　e_i ＝１ then Ａ＝ＡＢR^{-1}ｍｏｄＮ＋ｋＮ

(9)　｝

(10)　Ａ＝ＡR^{-1}ｍｏｄＮ＋ｋＮ

(11)　Ａ＝ＡｍｏｄＮ

(12)　ｏｕｔｐｕｔ　Ａ

【００６３】

この発明の他の実施例では、前記図２コプロセッサ２０９において、上記アルゴリズム５のステップ(4) 、ステップ(7) 、ステップ(8) 、ステップ(10)で示された「Ａ＝ＡＢR^{-1}ｍｏｄＮ＋ｋＮ」等に記述される「剰余乗算」を実行するようにされる。かかるコプロセッサ２０９は、後述するような演算回路と制御回路が含まれる。剰余乗算の入力値Ａ，Ｂ，Ｒ，Ｎ及び出力値Ａは専用レジスタ又はＲＡＭなどの記憶装置に保持される。

【００６４】

図２６には、この発明に用いられるコプロセッサの他の一実施例のブロック図が示されている。同図において３３は第１の積和演算器、３４は第２の積和演算器、３５は一次記憶値Ｔｅｍｐを保持するテンポラリレジスタ、３６は値Ａの格納に利用されるレジスタ、３７は値Ｂの格納に利用されるレジスタ、３８は値Ｎの格納に利用されるレジスタである。３９はＭｉ生成ロジック、４０はＭｉ生成ロジック３９で生成された値M_i を保持するラッチ、４１は「÷2^L 」を行うためのシフト回路である。

【００６５】

この実施例では、前記公報に詳細に説明されているようなブロック分割に基づいて演算「$(AB_i+M_iN)/2^L$」を実行するようにされる。先ず、第１の積和演算器３３は、レジスタ３５の値Ｔｅｍｐ、レジスタ３６の値Ａ、レジスタ３７の値B_iを入力として、積和演算「$Temp+A\cdot B_i$」を実行する。その演算結果は値Ｔｅｍｐ２として次段の第２の積和演算器３４へ送られる。上記値Ｔｅｍｐ２はｎ＋Ｌビット長の整数である。

【００６６】

一方、Ｍｉ生成ロジック３９は、Ｌビット長の数A_0，B_i，N_0を入力としてＬビットの整数M_iを生成し、この正数M_iはレジスタ４０に一時的に保持される。第２の積和演算器３４は、前記Ｔｅｍｐ２，Ｎ，M_iを入力として、積和演算「$Temp2+M_i\cdot N$」を実行する。ｎ＋Ｌビット長の演算結果の下位Ｌビットは全て０であり、これをシフタ４１によって消去して（すなわち2^Lで割って）、ｎビット長の結果が値Ｔｅｍｐとしてレジスタ３５に送られて保持される。

【００６７】

以上の動作をｎ／Ｌ回繰り返し実行すれば、演算「（ＡＢ＋ＭＮ）／Ｒ」が実現できる。これによれば、ｎビットの整数Ｍをあらかじめ計算して保持する必要はなく、Ｌビット長のM_iのみを積和演算器３３の計算中に求めてレジスタ４０に保持すればよく、値Ｍの計算時間の削除、および値Ｍを保持する記憶手段の規模を縮小することができる。さらに、積和演算器３３と積和演算器３４を直列的に接続して連続的に動作させることにより、ｎ＋Ｌビット長の中間結果Ｔｅｍｐ２を一時的に保持する記憶手段を特別に設けることも必要なくなる。

【００６８】

レジスタ３５～３８を積和演算器３３、３４にバス４３で接続される。したがって前記レジスタ３５～３８をＲＡＭ４２で構成することができる。これにより、半導体チップ上のレジスタ面積の低減が可能となる。また、この構成においては、特にデータバス４３によるデータ転送量が多いため、バス幅が大きくなって半導体チップの面積が大きくならないようにする必要が生じるが、図２６の実施例のように積和演算器３３と積和演算器３４を直列的に接続することにより、中

間結果Ｔｅｍｐ２をデータバスを用いて転送することが不要になるため、バスによるデータ転送量の低減を図ることができる。

【００６９】

この実施例のコプロセッサでは、第１の積和演算器３３でＴｅｍｐ＝０、第２の積和演算器３４でM_i・Ｎ＝０、さらにセレクタ４１による「$\div 2^L$」の動作を行なわないことにより、同図に示される演算手段を、「$A \cdot B_i$」のような多倍長乗算（小さな数B_iとその多倍長に相当する大きな数Ａとの乗算）を実行する回路として使用することができる。「$A \cdot B_i$」のような多倍長乗算演算は、上記アルゴリズムのステップ(2) の演算「R^2 ｍｏｄ　Ｎ」をマイクロプロセッサ２０１を用いて実行するときに適用されることにより、その演算の高速化を図ることができる。

【００７０】

図２７の「R^2 ｍｏｄＮ」の計算の概念図に示されていように、$R=2^n$、ｎ＝５１２とされ、Ｎは５１２ビット、R^2は最上位ビットだけが１で下位側１０２４ビット全てが０の値とされる。マイクロプロセッサで演算「R^2 ｍｏｄ　Ｎ」を行うとき、大きな数のR^2を同様に大きな数のＮで直接に除算するのは効率的でないから、被除数を最上位側から６４ビット単位のブロックとして把握し、また、除数を最上位側から３２ビット単位のブロックとして把握し、順次上位側のブロック同士を対象に除算を行い、それによって得られる値を商の概数として把握する。

【００７１】

同図において、例えばＱ（＝Ｄａ÷Ｎａ）を商の概数として把握する。概略的には、R^2の上位側に対して「Ｑ・Ｎａ」を減算し、その減算結果の上位側に対して「Ｑ・Ｎｂ」を減算する。「Ｑ・Ｎｂ」の減算結果に対して同様の処理を行い、更にその結果の対して同様の処理を繰り返すという手法によって、「R^2 ｍｏｄＮ」の結果を得ることができる。

【００７２】

実際にはその途上で、余剰ビットを消去するための減算処理が介在される。このとき、前記演算「Ｑ・Ｎｂ」の処理は、第１回目では３２ビットと４８０ビッ

トという大きな数の乗算処理とされる。しかもそのような大きな数の乗算処理は何回も繰り返される。このとき、前記図２６に示されるコプロセッサによって演算可能な前記「$A \cdot B_i$」のような多倍長乗算演算を利用することにより、換言すれば、そのような多倍長乗算演算をコプロセッサに負担させれば、上記アルゴリズム５におけるステップ２の演算「$R^2 \bmod N$」をマイクロプロセッサ２０１を用いて実行するとき、その演算処理の高速化を図ることができる。

【００７３】

前記のようなアルゴリズムにおける「$A = ABR^{-1} \bmod N$」の演算処理では、前記公報（特開平１０－２１０５７号）において詳述されているように、剰余乗算において、オーバーフロー有りのときには更に演算結果ＷからＮを減算Ｗ－Ｎするものであるため、オーバーフローの有無により演算時間や消費電流の違いが生じる。このため、前記のようなＩＣカードＬＳＩの消費電流を観測し、そのタイミングや統計的処理の結果からチップ内の動作を解析されてしまう可能性を持っている。

【００７４】

図２８には、この発明に係る暗号化処理用演算ユニットの一実施例の要部ブロック図が示されている。この実施例の暗号化処理用演算ユニットは、前記のようなＩＣカード等に搭載される１チップのマイクロコンピュータに含まれるコプロセッサに含まれる。

【００７５】

図２８において、前記図２６に示した第１と第２の積和演算器３３、３４を含む積和演算器により、前記$A^2 R^{-1} \bmod N$又は$ABR^{-1} \bmod N$の演算が行われ、その演算結果Ｗはテンポラリレジスタ Ｔｅｍｐに格納され、演算結果にオーバーフローが発生した場合には演算器からのオーバーフローフラグＯＶが制御論理のＯＶ格納レジスタに記憶される。そして、続いて上記剰余乗算の後にテンポラリレジスタ Ｔｅｍｐに格納された演算結果Ｗ－Ｎの減算が行われる。

【００７６】

上記オーバーフローフラグＯＶが有る時（論理１）には上記減算Ｗ－Ｎの結果は、テンポラリレジスタ Ｔｅｍｐに格納され、オーバーフローフラグＯＶが無い

時（論理０）には減算Ｗ－Ｎの結果は、テンポラリレジスタＴｅｍｐに格納されず、上記テンポラリレジスタＴｅｍｐ以外の適当な記憶回路、例えばレジスタＡに格納される。つまり、前記減算Ｗ－Ｎと、それにより形成された無効データを適当な記憶回路に格納する動作は、前記錯乱目的のダミー動作とされる。これにより、前記剰余乗算においてオーバーフローが生じないときでも、Ｗ－Ｎの減算及びその演算結果をレジスタに格納することに伴うＩＣカードの動作電流が常に発生し、オーバーフローの有無を外部より識別困難とすることができるものとなる。

【００７７】

上記の信号処理は、次のようなプログラムによって実施される。

```
Ｗ ← （ＡＢ＋ＭＮ）／Ｒ
Ｓｔｏｒｅ ＯＶ ｂｉｔ
ｉｆ ＯＶ ｔｈｅｎ
    Ｗ ← Ｗ－Ｎ （正規のオーバーフロー処理とＷへの書き込み）
Ｅｌｓｅ
    Ａ ← Ｗ－Ｎ （ダミーのオーバーフロー処理とＡへの書き込み）
Ｅｘｃｈａｎｇｅ Ｗ ａｎｄ Ａ
Ｏｕｔｐｕｔ Ａ
```

【００７８】

上記プログラムにおいて、Ｗはテンポラリレジスタ及びそのデータを表している。そして、オーバーフローフラグＯＶ無しのときに、テンポラリレジスタＴｅｍｐのアドレスをレジスタＡのアドレスを交換することにより、オーバーフローフラグＯＶの有り／無しに対応してＷ又はＡのデータが有効なデータとして出力される。この実施例では、Ｅｘｃｈａｎｇｅ Ｗ ａｎｄ Ａのようなアドレス交換によって、レジスタＡのアドレス指定によりテンポラリレジスタ（Ｗ）のデータを出力させるものである。

【００７９】

上記の信号処理は、次のようなプログラムに置き換えることができる。

```
Ｗ ← （ＡＢ＋ＮＭ）／Ｒ
```

Ｓｔｏｒｅ　ＯＶ　ｂｉｔ

Ａ　←　Ｗ－Ｎ　　（オーバーフロー処理とレジスタＡへの書き込み）

ｉｆ！　ＯＶ　ｔｈｅｎ

　　Ｅｘｃｈａｎｇｅ　Ｗ　ａｎｄ　Ａ

Ｅｌｓｅ　ｎｏｐ

Ｏｕｔｐｕｔ　Ａ

【００８０】

つまり、オーバーフローＯＶの有無に無条件でのオーバーフロー処理のための減算Ｗ－Ｎとその減算結果をＡレジスタの書き込みを行った後に、オーバーフローフラグＯＶが無ければ、Ｅｘｃｈａｎｇｅ　Ｗ　ａｎｄ　Ａのようにアドレス交換を行ってレジスタＡのアドレス指定によりテンポラリレジスタ（Ｗ）のデータを出力させ、無ければアドレスを交換することなくＡレジスタのデータＷ－Ｎを有効なデータとして出力させる。

【００８１】

上記の信号処理は、更に次のようなプログラムに置き換えることができる。

Ｗ　←　（ＡＢ＋ＮＭ）／Ｒ

Ｓｔｏｒｅ　ＯＶ　ｂｉｔ

Ｅｘｃｈａｎｇｅ　Ｗ　ａｎｄ　Ａ

Ｗ　←　Ａ－Ｎ　　（オーバーフロー処理とレジスタＡへの書き込み）

ｉｆ　ＯＶ　ｔｈｅｎ

　　Ｅｘｃｈａｎｇｅ　Ｗ　ａｎｄ　Ａ

Ｅｌｓｅ　ｎｏｐ

Ｏｕｔｐｕｔ　Ａ

【００８２】

つまり、オーバーフローＯＶの有無に無条件でのオーバーフロー処理のための減算Ｗ－Ａを行う前にＥｘｃｈａｎｇｅ　Ｗ　ａｎｄ　Ａのようにアドレス交換を行ってＡ－Ｎの減算、つまりはＷ－Ｎの減算を行ってテンポラリレジスタ（Ｗ）、つまりレジスタＡにデータを出力させる。そして、オーバーフローフラグＯＶが有れば、Ｅｘｃｈａｎｇｅ　Ｗ　ａｎｄ　Ａのように再度アドレス交換を行

い、レジスタＡのアドレス指定によりレジスタＡのデータを、オーバーフローフラグＯＶが無ければ前記のように交換したままレジスタＡのアドレス指定によりテンポラリレジスタ（Ｗ）のデータを出力させる。この構成では、レジスタへの書き込みを行う論理回路は、みかけ上テンポラリレジスタ（Ｗ）を書き込むような構成となり、レジスタＡへの書き込み用論理が不要となって回路の簡素化が可能になる。

【００８３】

前記テンポラリレジスタＴｅｍｐとレジスタＡのアドレス交換は、フラグ反転回路により実現できる。つまり、アドレスバスから供給されるアドレス信号のうち、例えば最下位ビットのような１ビットがテンポラリレジスタＴｅｍｐとレジスタＡとで異なるように設定しておき、フラグ反転回路により選択的にかかるビットを交換するだけで、テンポラリレジスタＴｅｍｐに割り当てられたアドレス指定によりレジスタＡを選択でき、逆にレジスタＡに割り当てられたアドレスによりテンポラリレジスタＴｅｍｐを選択することができる。

【００８４】

図２８の実施例において、２つのレジスタＴｅｍｐとＡを用い、オーバーフローフラグＯＶに対応して、常に一方（例えばテンポラリレジスタＴｅｍｐ）に有効データを格納させ、前記のようなアドレス交換によってレジスタＡを指定するアドレスにより有効なデータを出力させる。前記減算Ｗ－Ｎと、それにより形成された無効データを適当な記憶回路に格納する動作が錯乱目的のダミー動作とされる。これにより、前記剰余乗算においてオーバーフローが生じないときでも、Ｗ－Ｎの減算及びその演算結果をレジスタに格納することに伴うＩＣカードの動作電流が常に発生し、オーバーフローの有無を外部より識別困難とすることができるものとなる。

【００８５】

前記実施例のように積和演算器の前記のようなオーバーフローフラグに代えてボロー（Ｂｏｒｒｏｗ）フラグＢＲを利用するものであってもよい。つまり、前記$A^2\ R^{-1}$ｍｏｄＮ又はABR^{-1}ｍｏｄＮの演算結果Ｗはテンポラリレジスタ Ｔｅｍｐに格納し、Ｗ－Ｎの減算が行われときの演算器からのボローフラグＢＲを

記憶し、ボローフラグＢＲが有るときのみ、テンポラリレジスタＴｅｍｐとレジスタＡのアドレスを交換し、最終的にはレジスタＡのアドレス指定により有効なデータを読み出すようにするものであってもよい。

【００８６】

上記の信号処理は、次のようなプログラムにより実現できる。

```
W ← （AB＋NM）／R
A ←  W－N
Store BR bit
if BR then
    Exchange W and A
Else nop
Output A
```

【００８７】

図２９には、この発明に係る暗号化処理用演算ユニットの他の一実施例の要部ブロック図が示されている。この実施例の暗号化処理用演算ユニットも前記のようなＩＣカード等に搭載される１チップのマイクロコンピュータに含まれるコプロセッサに含まれる。この実施例では、データバスの信号と前記のような積和演算器の出力とのうちのいずれか一方を前記のようなＯＶフラグ格納レジスタに記憶されたオーバーフローフラグＯＶに従って出力させるセレクタが追加される。

【００８８】

上記剰余乗算の後Ｗ－Ｎの減算が行われ、この減算結果Ｗ－Ｎと演算のためにデータバス上に読み出されたＷの値がセレクタに入力され、オーバーフローフラグＯＶが有るときには減算結果Ｗ－Ｎが、オーバーフローフラグＯＶが無いときにはデータバスＷの値が選択され、この選択された値がレジスタＡに格納され、Ａが最終的に有効なデータとして出力されることにより、Ｗ－Ｎの減算とレジスタへの書込みに伴うＩＣカードやマイクロコンピュータの動作電流が常に発生し、オーバーフローの有無を外部より識別困難とされる。

【００８９】

図３０には、この発明に係る暗号化処理用演算ユニットの更に他の一実施例の

要部ブロック図が示されている。この実施例は、前記図２８の実施例においてレジスタブロックにレジスタＸが追加される。前記同様に剰余乗算の後Ｗ－Ｎの減算が行われ、オーバーフローフラグＯＶが有る時にはこの減算結果Ｗ－ＮがレジスタＡに、オーバーフローフラグＯＶが無い時にはダミー演算専用のレジスタＸに減算Ｗ－Ｎが書込まれる。この後に、オーバーフローフラグＯＶが無い時には、テンポラリレジスタ（Ｗ）とレジスタＡとのアドレスを交換して、最終的にレジスタＡを選択するアドレスにより有効なデータを出力させる。

【００９０】

上記の信号処理は、次のようなプログラムにより実現できる。

```
Ｗ ← （ＡＢ＋ＮＭ）／Ｒ
Ｓｔｏｒｅ　ＯＶ　ｂｉｔ
ｉｆ　ＯＶ　ｔｈｅｎ
　　Ａ ← Ｗ－Ｎ　（正規のオーバーフロー処理とＡへの書き込み）
Ｅｌｓｅ
　　Ｘ ← Ｗ－Ｎ　（ダミーのオーバーフロー処理とＸへの書き込み）
　　Ｅｘｃｈａｎｇｅ　Ｗ　ａｎｄ　Ａ
Ｏｕｔｐｕｔ　Ａ
```

【００９１】

上記の実施例から得られる作用効果は、下記の通りである。すなわち、

（１）　外部端子がリードライト装置と電気的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ったデータの入出力動作を含むＩＣカードにおいて、上記暗号化処理又は復号化処理に本来の処理動作に似た攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにすることによって、電流波形を利用した解読を困難にすることができるという効果が得られる。

【００９２】

（２）　上記に加えて、上記暗号化処理又は復号化処理は、ＲＳＡ暗号法などに応用可能なべき乗剰余乗算動作を含むようにすることにより、機密保護の強化を実現したＩＣカードを得ることができるという効果が得られる。

【００９３】

（３）　上記に加えて、上記べき乗剰余演算動作を中央処理装置からの指示を受けて動作する暗号処理用演算ユニットにより行わせることにより、高速なデータ処理を行なうようにすることができるという効果が得られる。

【００９４】

（４）　上記に加えて、上記暗号化処理用演算ユニットの動作として、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、$A=A^2 \bmod N$とＡ＝ＡＢｍｏｄＮの演算を交互行ない、かかる演算においてＹの上位から１ビットずつみて、論理０であれば上記$A^2 \bmod N$の演算結果を有効なデータとして記憶回路に取り込み、論理１であれば上記$A^2 \bmod N$とＡＢｍｏｄＮの演算結果を有効なデータとして記憶回路に取り込むものとし、上記論理０のときのＡ＝ＡＢｍｏｄＮの演算動作を上記攪乱目的のダミー処理動作とすることにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【００９５】

（５）　上記に加えて、上記記憶回路をリードライトバッファを通してデータの入出力が行なわれる複数のレジスタからなるレジスタブロックを用い、上記Ｙの特定ビットe_iの論理１又は０によってゲート回路を制御し、所定のレジスタに供給されるライトストローブ信号の伝達を制御して、上記演算結果のうち有効なデータのみがリードライトバッファを通して上記所定のレジスタに格納することにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【００９６】

（６）　上記に加えて、上記記憶回路をリードライトバッファを通してデータの入出力が行なわれる複数のレジスタとからなるレジスタブロックを用い、上記Ｙの特定ビットe_iの論理１又は０によってゲート回路を制御し、上記リードライトバッファに供給されるライトストローブ信号の伝達を制御して、上記演算結果のうち有効なデータのみがリードライトバッファを通して上記所定のレジスタに格納することにより、暗号処理を行いつつ電流波形を利用した解読を困難にする

ことができるという効果が得られる。

【００９７】

（７）　上記に加えて、上記記憶回路をリードライトバッファを通してデータの入出力が行なわれる複数のレジスタ及びダミーレジスタとからなるレジスタブロックを用い、上記リードライトバッファと上記ダミーレジスタ及び複数のレジスタとの間セレクタを設けて上記Ｙの特定ビットe_iの論理１又は０によって制御して、リードライトバッファに書き込まれた演算結果のうち有効なデータを所定のレジスタに格納し、無効なデータが上記ダミーレジスタに格納することにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にすることができるという効果が得られる。

【００９８】

（８）　上記に加えて、上記暗号化処理用演算ユニットの動作として、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、$A=A^2 \bmod N$と$A=AB \bmod N$の演算を交互行ない、かかる演算においてＹの上位から１ビットずつみて、論理０であれば上記$A^2 \bmod N$の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込み、論理１であれば上記$A^2 \bmod N$と$AB \bmod N$の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込むものであり、上記$A=A^2 \bmod N$の演算結果が出力されてから上記$A=AB \bmod N$の演算が開始されまるまでの間も上記$A=A^2 \bmod N$の動作を継続し、$A=AB \bmod N$の演算結果が出力されてからＹのビットの変更判定処理を含めて次のビットに対応した$A^2 \bmod N$の演算が開始されるまでの間も上記$A=AB \bmod N$の動作を継続することにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にすることができるという効果が得られる。

【００９９】

（９）　上記に加えて、上記暗号化処理用演算ユニットの動作として、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、$A=A^2 \bmod N$と$A=AB \bmod N$の演算とそれぞれに対してオーバーフロー演算行ない、かかる演算においてＹの上位から１ビットずつみて、論理０であれば上記$A^2 \bmod N$の演算結果を有効なデータとして記憶回路に取り込み、論理１であれば上記$A^2 \bmod N$と

ＡＢｍｏｄＮの演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理０のときのＡ＝ＡＢｍｏｄＮの演算動作と、各演算動作での不要なオーバーフロー演算を上記攪乱目的のダミー処理動作とすることにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にすることができるという効果が得られる。

【０１００】

（１０）　外部端子がリードライト装置と電気的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ってデータの入出力動作が行われるＩＣカードに、上記暗号化処理又は復号化処理に攪乱目的のダミー演算を含ませて内部回路の動作タイミング及び動作電流に不規則性を持たせることにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にしたＩＣカードを得ることができるという効果が得られる。

【０１０１】

（１１）　外部端子がリードライト装置と電気的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ってデータの入出力動作が行われるＩＣカードに、上記暗号化処理又は復号化処理における各演算の間隔に攪乱目的のダミーサイクルを含ませて内部回路の動作タイミング及び動作電流に不規則性を持たせることにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にしたＩＣカードを得ることができるという効果が得られる。

【０１０２】

（１２）　暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにすることにより、モジュール化されたマイクロコンピュータに対する電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１０３】

（１３）　上記に加えて、上記マイクロコンピュータのモジュール構成を１つの

半導体基板上において形成することにより、小型化を図りつつ電流波形以外の直接的なプログラム又はデータ等の解読も防止することができるという効果が得られる。

【０１０４】

（１４）　上記に加えて、上記マイクロコンピュータの暗号化処理又は復号化処理を、ＲＳＡ暗号法などに応用可能なべき乗剰余乗算動作を含むものとし、上記べき乗剰余乗算動作を中央処理装置からの指示を受けて動作する暗号処理用演算ユニットにより行なうようにすることにより、高速な暗号処理動作を行なうようにすることができるという効果が得られる。

【０１０５】

（１５）　上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットの動作として、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、Ａ＝A^2 ｍｏｄＮとＡ＝ＡＢｍｏｄＮの演算を行ない、かかる演算においてＹの上位から１ビットずつみて、論理０であれば上記A^2 ｍｏｄＮの演算結果を有効なデータとして記憶回路に取り込み、論理１であれば上記A^2 ｍｏｄＮとＡＢｍｏｄＮの演算結果を有効なデータとして記憶回路に取り込むものとし、上記論理０のときのＡ＝ＡＢｍｏｄＮの演算動作を上記攪乱目的のダミー処理動作とすることにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１０６】

（１６）　上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットの動作として、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、Ａ＝A^2 ｍｏｄＮとＡ＝ＡＢｍｏｄＮの演算を行ない、かかる演算においてＹの上位から１ビットずつみて、論理０であれば上記A^2 ｍｏｄＮの演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込み、論理１であれば上記A^2 ｍｏｄＮとＡＢｍｏｄＮの演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込むものであり、上記Ａ＝A^2 ｍｏｄＮの演算結果が出力されてから上記Ａ＝ＡＢｍｏｄＮの演算が開始されまるまでの間も上記Ａ＝A^2 ｍｏｄＮの動作を継続し、Ａ＝ＡＢｍｏｄＮの演算結果が出力されてからＹのビットの

変更判定処理を含めて次のビットに対応したA^2 ｍｏｄＮの演算が開始されるまでの間も上記Ａ＝ＡＢｍｏｄＮの動作を継続することにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

（１７）　上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットの動作として、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、Ａ＝X^Y ｍｏｄＮとＡ＝ＡＢｍｏｄＮの演算とそれぞれに対してオーバーフロー演算を行ない、かかる演算においてＹの上位から１ビットずつみて、論理０であれば上記A^2 ｍｏｄＮの演算結果を有効なデータとして記憶回路に取り込み、論理１であれば上記A^2 ｍｏｄＮとＡＢｍｏｄＮの演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理０のときのＡ＝ＡＢｍｏｄＮの演算動作と、各演算動作での不要なオーバーフロー演算を上記攪乱目的のダミー処理動作とすることにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１０７】

（１８）　上記に加えて、上記暗号化処理用演算ユニットにより、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、Ｙのビットの値に応じて、Ａ＝A^2R^{-1}ｍｏｄＮ、Ａ＝ＡBR^{-1}ｍｏｄＮの演算行うとともに、演算結果にオーバーフローが発生した場合にはさらに上記演算結果ＷからＮの減算Ｗ－Ｎを行なう正規動作と、各々の演算結果にオーバーフローが発生しない場合でも上記減算Ｗ－Ｎに対応した演算による無効データを生成する攪乱目的のダミー動作を行い、上記オーバーフローの有無に対応して有効なデータを出力させることにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１０８】

（１９）　上記に加えて、上記A^2 R^{-1}ｍｏｄＮ又はＡBR^{-1}ｍｏｄＮの演算結果Ｗを第１の記憶回路に格納し、演算器のオーバーフローフラグＯＶの有無を記憶し、上記剰余乗算の後に上記第１記憶回路の演算結果ＷからＮの減算Ｗ－Ｎを行い、その演算結果を上記オーバーフローフラグＯＶが有る時には上記第１の記憶回路に格納し、オーバーフローフラグＯＶが無い時には上記第１記憶回路とは

異なる第２の記憶回路に上記錯乱目的のダミー動作として格納し、上記第１の記憶回路の演算結果を有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１０９】

（２０）　上記に加えて、上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗを第１の記憶回路に格納し、演算器のオーバーフローフラグＯＶの有無を記憶し、上記剰余乗算の後に上記第１の記憶回路の演算結果ＷからＮの減算Ｗ－Ｎを行い、オーバーフローフラグＯＶが有るときに上記演算結果Ｗ－Ｎをセレクタにより選択され、オーバーフローフラグＯＶが無いときには上記第１記憶回路の演算結果Ｗをセレクタにより選択して第２の記憶回路に格納することにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１１０】

（２１）　上記に加えて、上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗを第１の記憶回路に格納し、演算器のオーバーフローフラグＯＶの有無を記憶し、上記剰余乗算の後に上記第１の記憶回路の演算結果ＷからＮの減算Ｗ－Ｎを行い、オーバーフローフラグＯＶが有るときには減算Ｗ－Ｎを第２の記憶回路に記憶し、オーバーフローフラグＯＶが無いときには減算Ｗ－Ｎを第３の記憶回路に記憶し、オーバーフローフラグＯＶが有るときには上記第２の記憶回路のデータが有効なデータとして出力し、オーバーフローフラグＯＶが無いときには上記第１の記憶回路のデータが有効なデータとして出力することにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１１１】

（２２）　上記に加えて、上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗを第１の記憶回路に格納し、演算器のオーバーフローフラグＯＶの有無を記憶し、上記剰余乗算の後に上記第１の記憶回路の演算結果ＷからＮの減算結果Ｗ－Ｎを第２の記憶回路に格納し、オーバーフローフラグＯＶが無いとき第１の記

憶回路と第２の記憶回路を選択する最下位アドレスを逆転させて、上記第２の記憶回路を選択するアドレスにより第１の記憶回路を選択して有効なデータとして出力させ、オーバーフローフラグＯＶが有るとき第１の記憶回路と第２の記憶回路を選択する最下位アドレスをそのままにして第２の記憶回路の演算結果を有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１１２】

（２３）　上記に加えて、上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗを第１の記憶回路に格納し、演算器のオーバーフローフラグＯＶの有無を記憶し、上記剰余乗算の後に上記第１の記憶回路と第２の記憶回路のアドレスを交換し、第２の記憶回路を選択するアドレスにより選択される演算結果値ＷからＮの減算Ｗ－Ｎが行われて第１の記憶回路を選択するアドレスにより選択される第２の記憶回路に減算結果Ｗ－Ｎを格納し、オーバーフローフラグＯＶが有るときにのみ上記アドレスを再度交換し、第１の記憶回路を選択するアドレスにより選択される第１又は第２の記憶回路のデータを有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１１３】

（２４）　上記に加えて、上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗを第１の記憶回路に格納し、上記剰余乗算の後に上記第１の記憶回路の演算結果値ＷからＮの減算Ｗ－Ｎを行って第２の記憶回路に格納し、このＷ－Ｎの減算が行われた時の演算器のボローフラグＢＲを記憶し、ボローフラグＢＲが有るときには、第１の記憶回路と第２の記憶回路を選択する最下位アドレスを逆転させて、上記第２の記憶回路を選択するアドレスにより第１の記憶回路の演算結果Ｗを出力し、ボローフラグＢＲが無いときには、第１の記憶回路と第２の記憶回路を選択する最下位アドレスをそのままして、上記第２の記憶回路を選択するアドレスにより第２の記憶回路の演算結果Ｗ－Ｎを出力させることにより、暗号化

処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１１４】

（２５）　上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットとして、入力されたＸ、Ｙ及びＮを受け、Ａ＝１、Ｂ＝Ｘとして、Ｙのビットの値に応じて、$A=A^2\ R^{-1} \mathrm{mod} N$、$A=ABR^{-1} \mathrm{mod} N$の演算を行い、演算結果にオーバーフローが発生した場合にはさらに上記演算結果ＷからＮの減算Ｗ－Ｎを行なう正規動作と、各々の演算結果にオーバーフローが発生しない場合でも上記減算Ｗ－Ｎに対応した演算による無効データを生成する攪乱目的のダミー動作を行い、上記オーバーフローの有無に対応して有効なデータを出力させることより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１１５】

（２６）　上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記$A^2\ R^{-1} \mathrm{mod} N$又は$ABR^{-1} \mathrm{mod} N$の演算結果Ｗは第１の記憶回路に格納し、演算器からのオーバーフローフラグＯＶの有無を記憶し、上記剰余乗算の後に上記第１記憶回路の演算結果ＷからＮの減算Ｗ－Ｎを行ない、演算結果を上記オーバーフローフラグＯＶが有る時には上記第１の記憶回路に格納し、オーバーフローフラグＯＶが無い時には上記第１記憶回路とは異なる第２の記憶回路に上記錯乱目的のダミー動作として書き込み、上記第１の記憶回路の演算結果が有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１１６】

（２７）　上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記$A^2\ R^{-1} \mathrm{mod} N$又は$ABR^{-1} \mathrm{mod} N$の演算結果Ｗを第１の記憶回路に格納し、演算器のオーバーフローフラグＯＶの有無を記憶し、上記剰

余乗算の後に上記第１の記憶回路の演算結果ＷからＮの減算Ｗ－Ｎを行ってオーバーフローフラグＯＶが有るときにセレクタにより上記演算結果Ｗ－Ｎを選択し、オーバーフローフラグＯＶが無いときにはセレクタにより上記第１記憶回路の演算結果Ｗを選択して第２の記憶回路に格納して有効なデータとして出力することにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１１７】

（２８）　上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記$A^2 R^{-1} mod N$又は$ABR^{-1} mod N$の演算結果Ｗを第１の記憶回路に格納し、演算器のオーバーフローフラグＯＶの有無を記憶し、上記剰余乗算の後に上記第１の記憶回路の演算結果ＷからＮの減算Ｗ－Ｎを行ってオーバーフローフラグＯＶが有るときには減算結果Ｗ－Ｎを第２の記憶回路に記憶し、オーバーフローフラグＯＶが無いときには減算結果Ｗ－Ｎを第３の記憶回路に記憶し、オーバーフローフラグＯＶが有るときには上記第２の記憶回路のデータが有効なデータとして出力し、オーバーフローフラグＯＶが無いときには上記第１の記憶回路のデータを有効なデータとして出力することにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１１８】

（２９）　上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記$A^2 R^{-1} mod N$又は$ABR^{-1} mod N$の演算結果Ｗを第１の記憶回路に格納し、演算器のオーバーフローフラグＯＶの有無を記憶し、上記剰余乗算の後に上記第１の記憶回路の演算結果ＷからＮの減算結果Ｗ－Ｎを第２の記憶回路に格納し、オーバーフローフラグＯＶが無いとき第１の記憶回路と第２の記憶回路を選択する最下位アドレスを逆転させて、上記第２の記憶回路を選択するアドレスにより第１の記憶回路を選択して有効なデータとして出力させ、オーバーフローフラグＯＶが有るとき第１の記憶回路と第２の記憶回路を選択する最下位アドレスをそのままにして第２の記憶回路の演算結果を有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタ

への書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１１９】

（３０）　上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗを第１の記憶回路に格納し、演算器のオーバーフローフラグＯＶの有無を記憶し、上記剰余乗算の後に上記第１の記憶回路と第２の記憶回路のアドレスを交換し、第２の記憶回路を選択するアドレスにより選択される演算結果値ＷからＮの減算Ｗ－Ｎを行って第１の記憶回路を選択するアドレスにより選択される第２の記憶回路に減算結果Ｗ－Ｎを格納し、オーバーフローフラグＯＶが有るときにのみ上記アドレスを再度交換し、第１の記憶回路を選択するアドレスにより選択される第１又は第２の記憶回路のデータを有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１２０】

（３１）　上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記$A^2\ R^{-1}$ｍｏｄＮ又はＡＢR^{-1}ｍｏｄＮの演算結果Ｗを第１の記憶回路に格納し、上記剰余乗算の後に上記第１の記憶回路の演算結果値ＷからＮの減算Ｗ－Ｎを行って第２の記憶回路に格納し、このＷ－Ｎの減算が行われた時の演算器からボローフラグＢＲを記憶し、ボローフラグＢＲが有るときには、第１の記憶回路と第２の記憶回路を選択する最下位アドレスを逆転させて、上記第２の記憶回路を選択するアドレスにより第１の記憶回路の演算結果Ｗを出力し、ボローフラグＢＲが無いときには、第１の記憶回路と第２の記憶回路を選択する最下位アドレスをそのまましまして、上記第２の記憶回路を選択するアドレスにより第２の記憶回路の演算結果Ｗ－Ｎを出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【０１２１】

以上本発明者よりなされた発明を実施例に基づき具体的に説明したが、本願発明は前記実施例に限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能であることはいうまでもない。例えば、ＩＣカードには、１つの半導体集積回路装置を搭載するもの他、複数の半導体集積回路装置が搭載されるものであってもよい。マイクロコンピュータは、１つの半導体集積回路装置に形成されるもの他、ＣＰＵとその周辺回路が複数チップで構成されて、１つのモジュール基板に搭載されてなるものであってもよい。

【０１２２】

演算処理は前記のような暗号処理を行なうべき乗剰余乗算法の他に、図２５図に示したフロチャート図のように演算Ａと演算Ｂを持ち、演算Ａの結果により演算Ｂを行なうか否かの分岐を持つような演算処理、あるいは演算動作てのオーバーフローの有無に対応して、次の演算処理が選択的に追加される場合に等に広く利用することができる。つまり、演算Ａの次に演算Ｂを実行し、演算Ａの結果から演算Ｂが不要なら、その演算結果を無効にするような演算処理を行なえば、前記のような暗号処理以外の機密動作を必要とするデータ処理のハッキング対策として有益なものとなる。

【０１２３】

上記マイクロコンピュータは、データ処理装置とかかるデータ処理装置によるデータ処理手順が書き込まれたＲＯＭを含んで記データ処理手順に従ってデータの入出力動作が行われるものであれば何であってもよい。例えば、前記のようなＩＣカード用チップの他に、ゲーム用等の１チップマイクロコンピュータ等のように機密保護の必要な各種マイクロコンピュータに広く適用できるものである。この発明は、機密保護を必要とする各種ＩＣカード及びマイクロコンピュータに広く利用できる。

【０１２４】

【発明の効果】

本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば、下記の通りである。すなわち、外部端子がリードライト装置と電気的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号

化処理を伴ったデータの入出力動作を含むＩＣカードにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにすることによって、電流波形を利用した解読を困難にすることができる。

【０１２５】

暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにすることにより、モジュール化されたマイクロコンピュータに対する電流波形を利用した解読を困難にすることができる。

【図面の簡単な説明】

【図１】

この発明が適用されるＩＣカードの一実施例を示す外観図である。

【図２】

この発明に係るＩＣカードに搭載されるＩＣカード用チップの一実施例を示す概略ブロック図である。

【図３】

この発明に係るコプロセッサの一実施例の動作を説明するためのタイミング図である。

【図４】

図３のコプロセッサの動作を説明するためのフローチャート図である。

【図５】

図３のコプロセッサの一実施例を示すブロック図である。

【図６】

図３に示したコプロセッサの動作を実現するための一実施例を示すブロック図である。

【図７】

図３のコプロセッサの他の一実施例を示すブロック図である。

【図８】

図３のコプロセッサの他の一実施例を示すブロック図である。

【図９】

この発明に係るコプロセッサの他の一実施例の動作を説明するための構成図である。

【図１０】

図９に示したコプロセッサの動作を実現するための一実施例を示すブロック図である。

【図１１】

この発明に係るコプロセッサの他の一実施例の動作を説明するためのタイミング図である。

【図１２】

この発明に係るコプロセッサの他の一実施例の動作を説明するためのフローチャート図である。

【図１３】

この発明に係るコプロセッサの他の一実施例の動作の詳細を説明するためのタイミング図である。

【図１４】

図１１ないし図１３に示したコプロセッサの動作を実現するための一実施例を示すブロック図である。

【図１５】

この発明に係るコプロセッサの更に他の一実施例の動作を説明するためのタイミング図である。

【図１６】

この発明に係るコプロセッサの演算動作の他の一実施例を示すフローチャート図である。

【図１７】

この発明に係るコプロセッサの他の一実施例を示すブロック図である。

【図１８】

この発明に係るコプロセッサの他の一実施例を示すブロック図である。

【図１９】

この発明に係るコプロセッサの他の一実施例を示すブロック図である。

【図２０】

この発明に係るＩＣカード用チップの他の一実施例を示す要部ブロック図である。

【図２１】

図２０のカウンタの一実施例を示すブロック図である。

【図２２】

図２０のＩＣカード用チップの動作の一例を示すタイミング図である。

【図２３】

この発明に係るＩＣカード用チップの更に他の一実施例を示す要部ブロック図である。

【図２４】

図２３のＩＣカード用チップの動作の一例を示すタイミング図である。

【図２５】

この発明が適用可能な演算動作を説明するためのフローチャート図である。

【図２６】

この発明に用いられるコプロセッサの他の一実施例を示すブロック図である。

【図２７】

この発明における「R^2 ｍｏｄＮ」の計算方法を示す概念図である。

【図２８】

この発明に係る暗号化処理用演算ユニットの一実施例を示す要部ブロック図である。

【図２９】

この発明に係る暗号化処理用演算ユニットの他の一実施例を示す要部ブロック図である。

【図３０】

この発明に係る暗号化処理用演算ユニットの更に他の一実施例を示す要部ブロック図である。

【符号の説明】

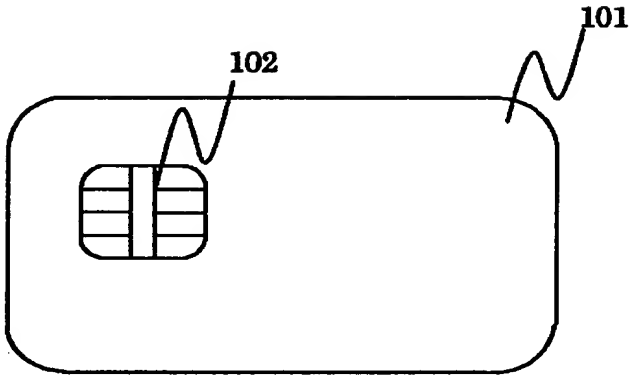
２０１…中央処理装置（ＣＰＵ）、２０２…Ｉ／Ｏポート、２０３…アドレスバス、２０４…データバス、２０５…クロック生成回路、２０６…ＲＯＭ、２０７…ＲＡＭ、２０８…ＥＥＰＲＯＭ、２０９…コプロセッサ（暗号化処理用演算ユニット）、

ＣＤＡ、ＣＤＢ、ＣＤＮ、ＣＤＷ…レジスタ。

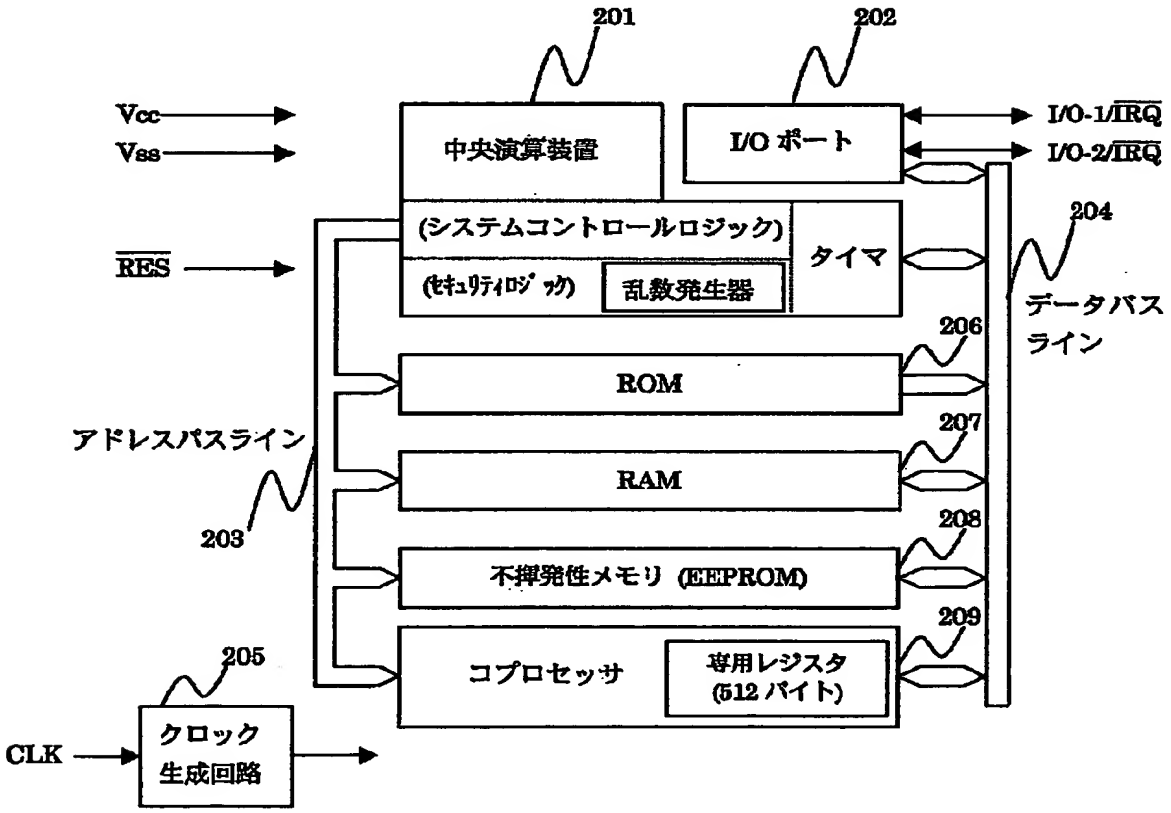
３３，３４…積和演算器、３５…テンポラリレジスタ、３６～３８…レジスタ、３９…Ｍｉ生成ロジック、４０…Ｍｉを保持するラッチ（レジスタ）、４１…シフタ、４２…ＲＡＭ、４３…データバス。

【書類名】　　図面

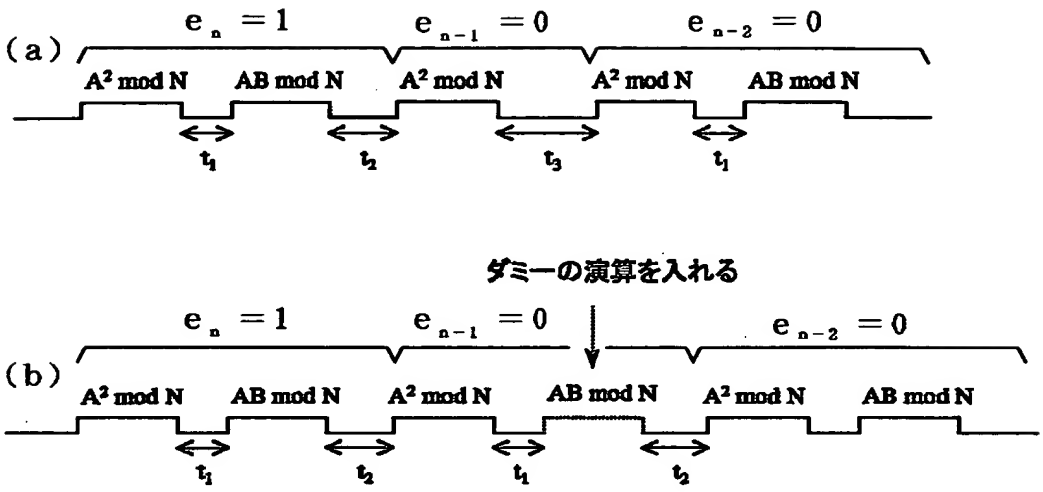
【図１】


101
102

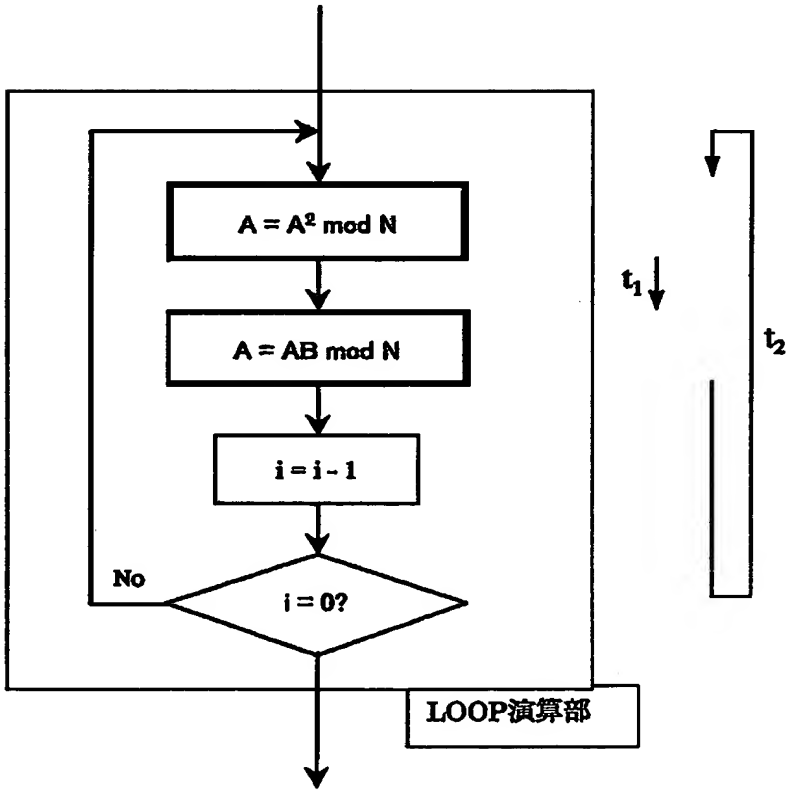
【図２】


201
202
Vcc
Vss
中央演算装置
I/O ポート
I/O-1/IRQ
I/O-2/IRQ
(システムコントロールロジック)
タイマ
204
RES
(ｾｷｭﾘﾃｨﾛｼﾞｯｸ)
乱数発生器
データバス
ライン
206
ROM
アドレスバスライン
207
RAM
203
208
不揮発性メモリ (EEPROM)
209
コプロセッサ
専用レジスタ
(512 バイト)
205
CLK
クロック
生成回路

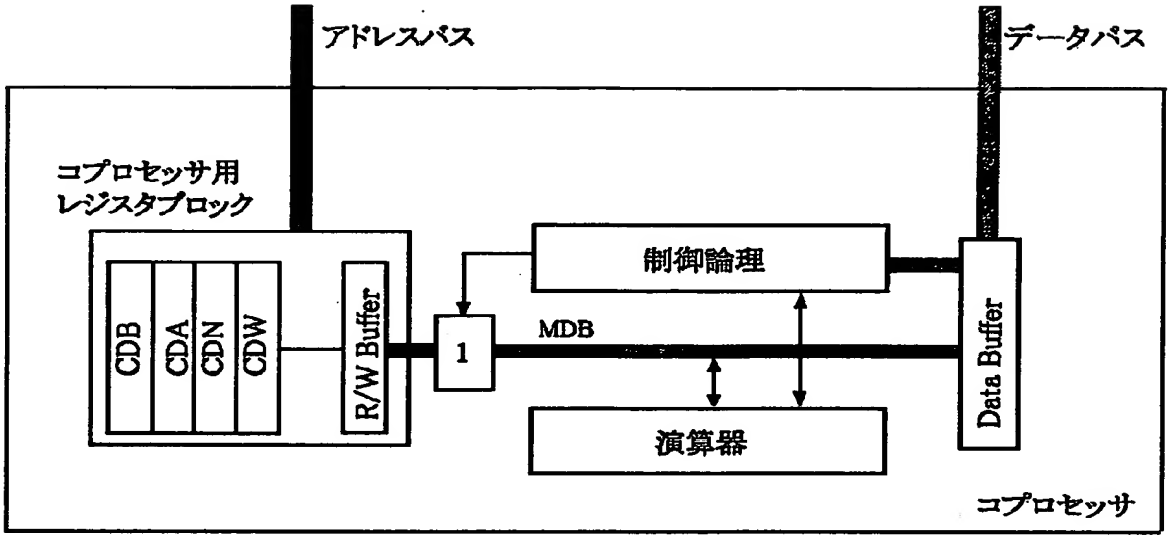
【図３】


(a)
$e_n = 1$
$e_{n-1} = 0$
$e_{n-2} = 0$
A^2 mod N
AB mod N
A^2 mod N
A^2 mod N
AB mod N
t_1
t_2
t_3
t_1
ダミーの演算を入れる
(b)
$e_n = 1$
$e_{n-1} = 0$
$e_{n-2} = 0$
A^2 mod N
AB mod N
A^2 mod N
AB mod N
A^2 mod N
AB mod N
t_1
t_2
t_1
t_2

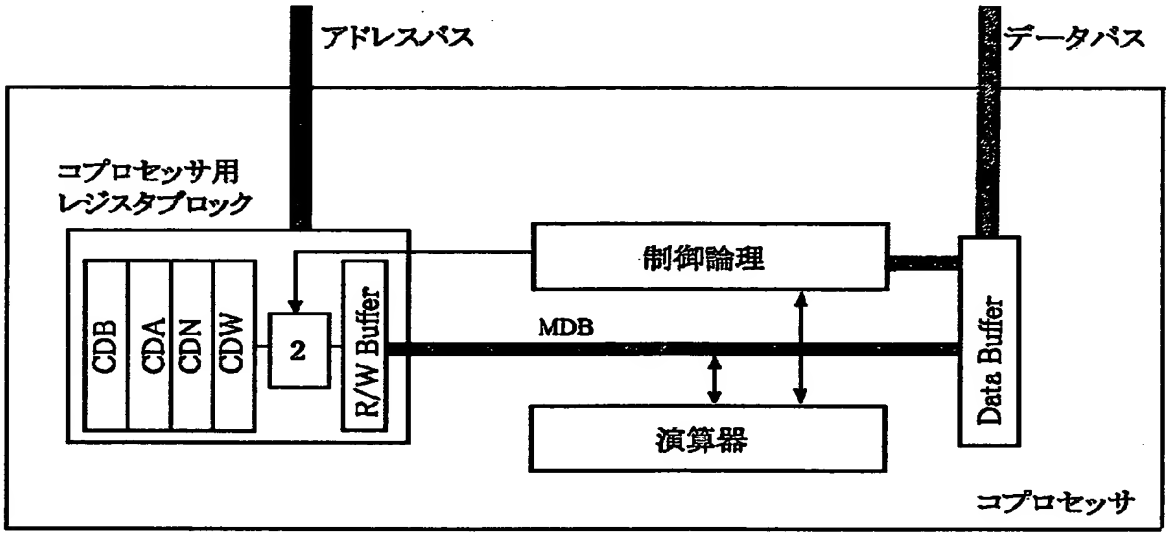
【図４】


A = A² mod N
A = AB mod N
i = i - 1
No
i = 0?
LOOP演算部
t_1
t_2

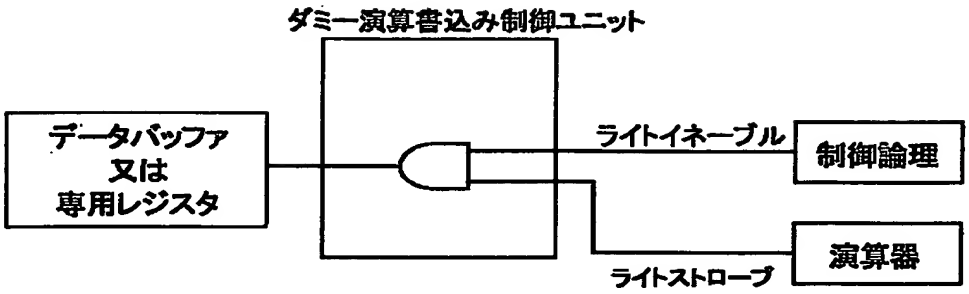
【図５】


アドレスバス
データバス
コプロセッサ用
レジスタブロック
CDB
CDA
CDN
CDW
R/W Buffer
1
制御論理
MDB
Data Buffer
演算器
コプロセッサ

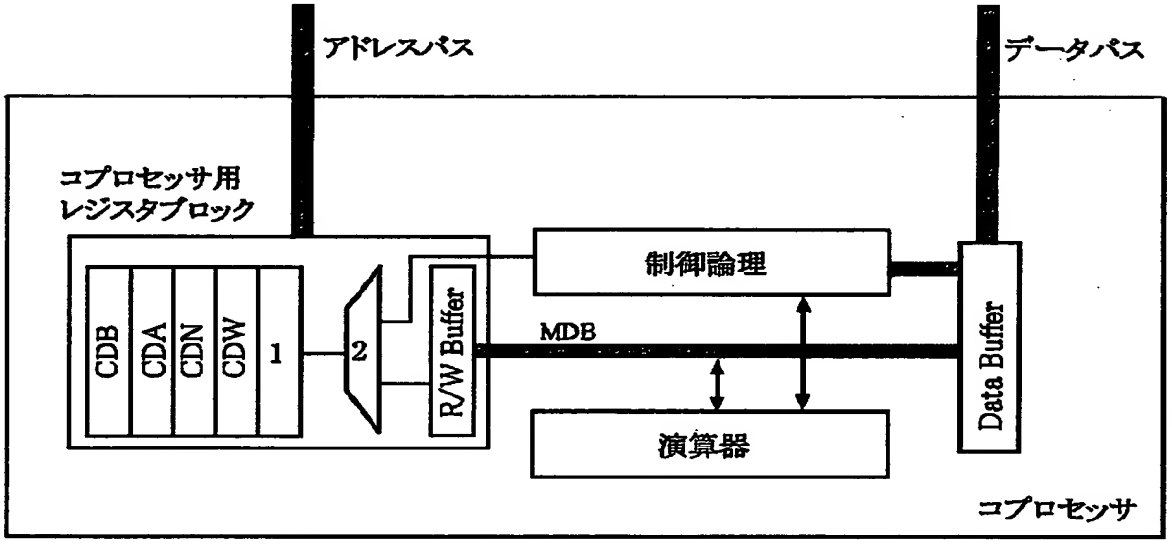
【図６】


アドレスバス
データバス
コプロセッサ用
レジスタブロック
CDB
CDA
CDN
CDW
2
R/W Buffer
制御論理
MDB
Data Buffer
演算器
コプロセッサ

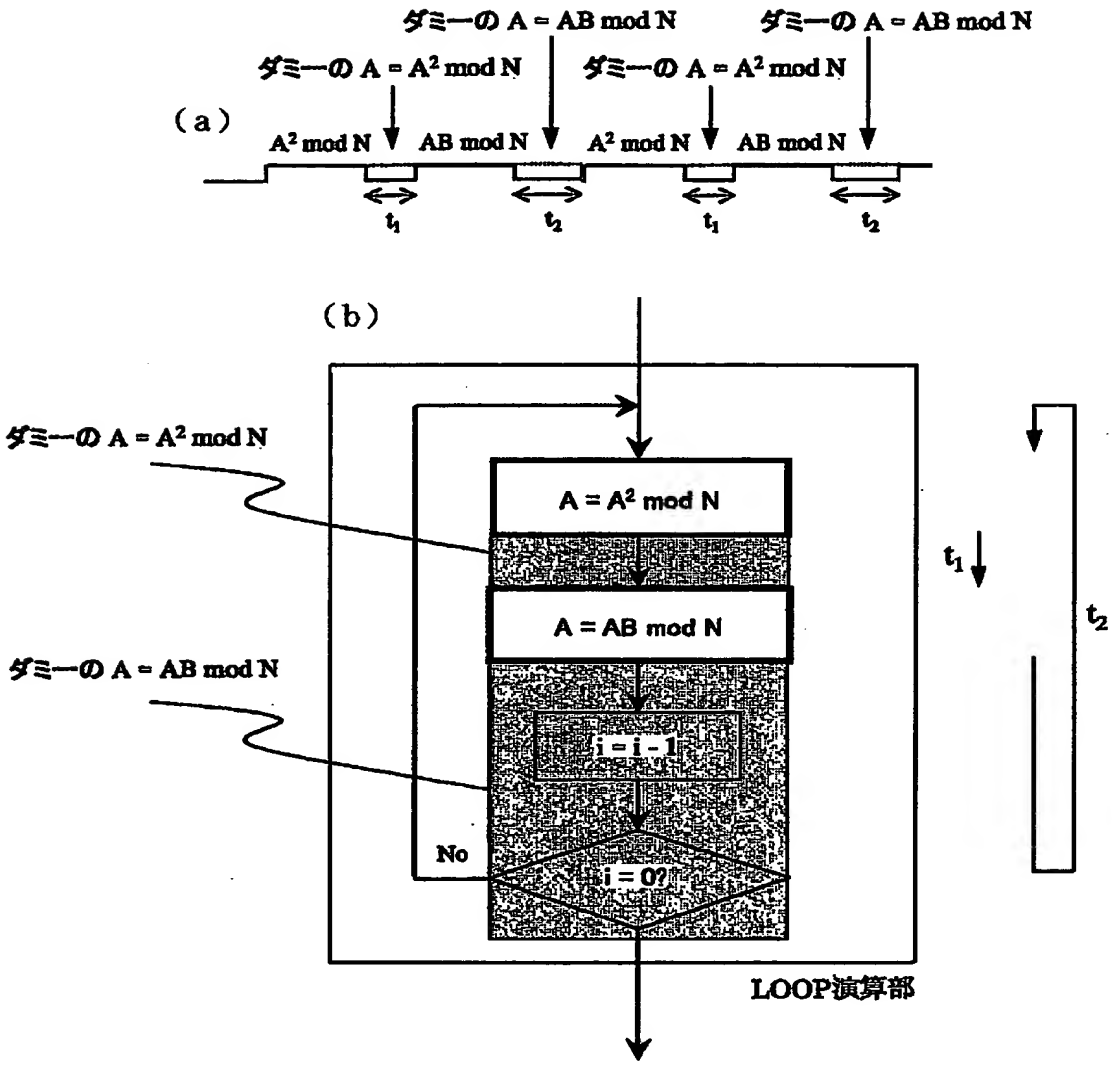
【図７】


ダミー演算書込み制御ユニット
データバッファ
又は
専用レジスタ
ライトイネーブル
制御論理
演算器
ライトストローブ

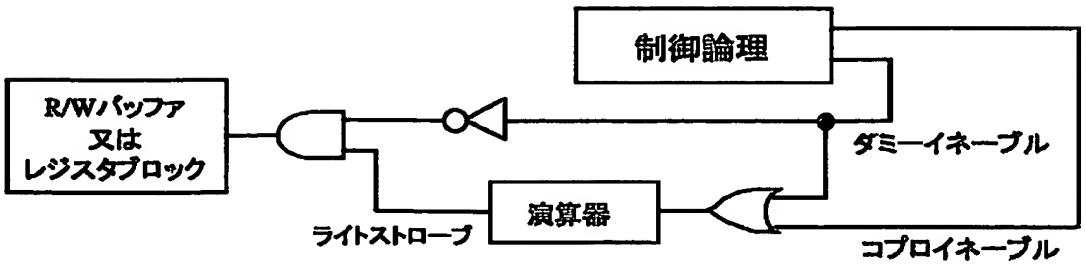
【図８】


アドレスバス
データバス
コプロセッサ用
レジスタブロック
CDB
CDA
CDN
CDW
1
2
R/W Buffer
制御論理
MDB
Data Buffer
演算器
コプロセッサ

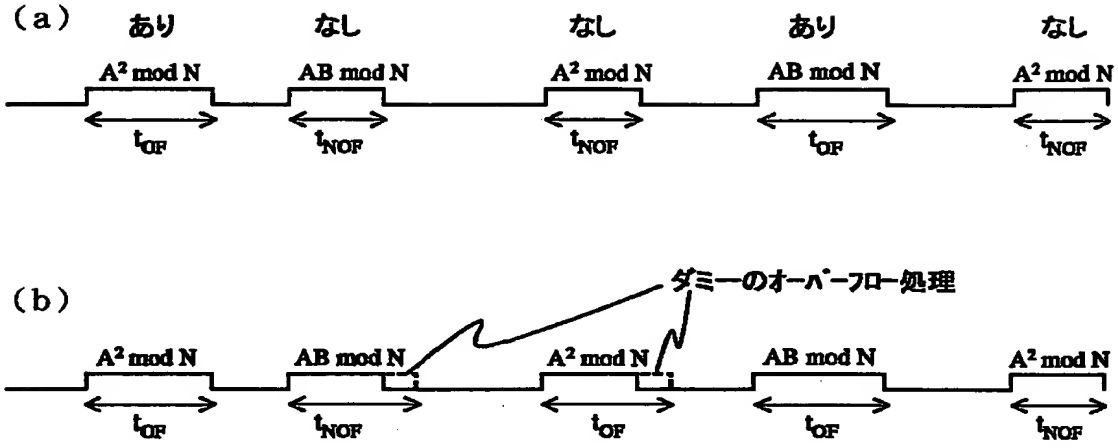
【図９】


ダミーの A = AB mod N
ダミーの A = A² mod N
ダミーの A = A² mod N
ダミーの A = AB mod N
(a)
A² mod N
AB mod N
A² mod N
AB mod N
t1
t2
t1
t2
(b)
ダミーの A = A² mod N
ダミーの A = AB mod N
A = A² mod N
A = AB mod N
i = i - 1
No
i = 0?
t1
t2
LOOP演算部

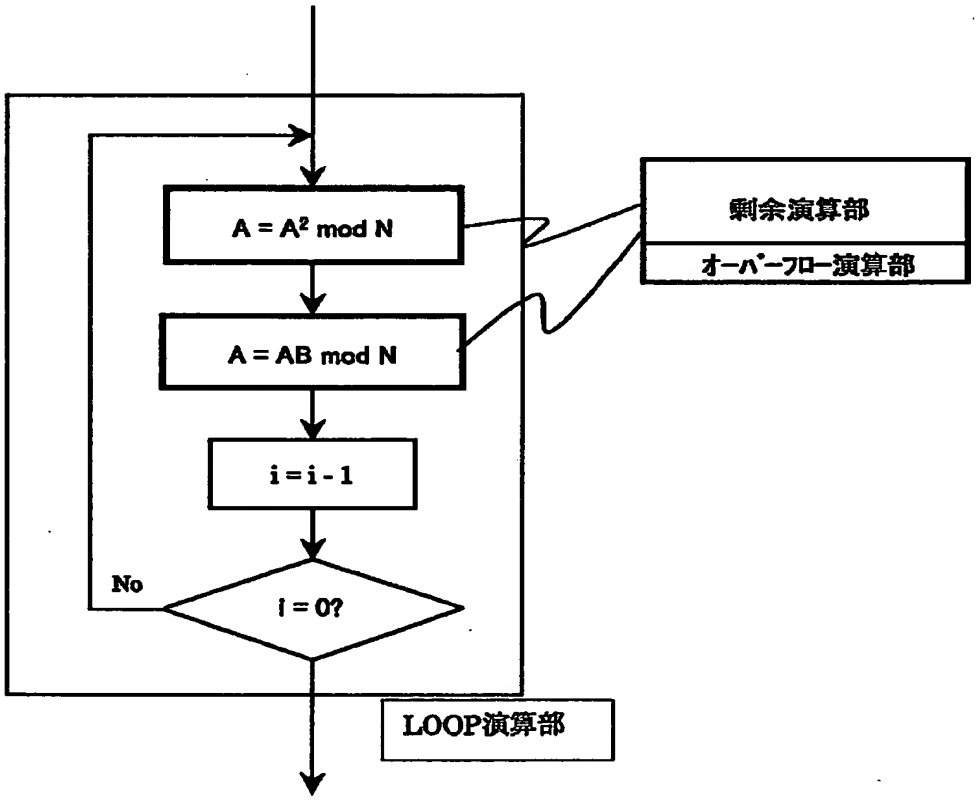
【図１０】


制御論理
R/Wバッファ
又は
レジスタブロック
ダミーイネーブル
演算器
ライトストローブ
コプロイネーブル

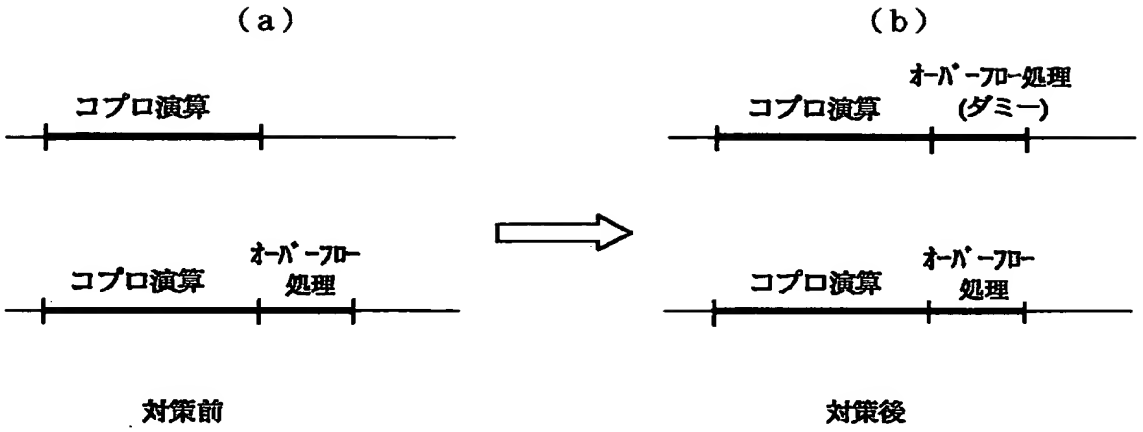
【図１１】


(a)
あり
なし
なし
あり
なし
A² mod N
AB mod N
A² mod N
AB mod N
A² mod N
tOF
tNOF
tNOF
tOF
tNOF
(b)
ダミーのオーバーフロー処理
A² mod N
AB mod N
A² mod N
AB mod N
A² mod N
tOF
tNOF
tOF
tOF
tNOF

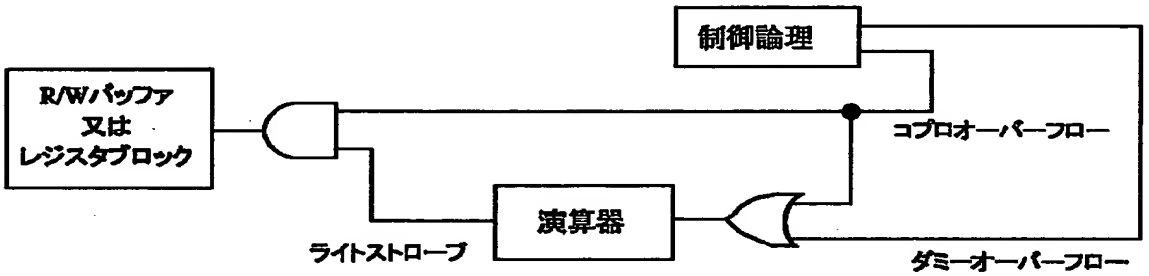
【図１２】


A = A² mod N
A = AB mod N
i = i - 1
No
i = 0?
剰余演算部
オーバーフロー演算部
LOOP演算部

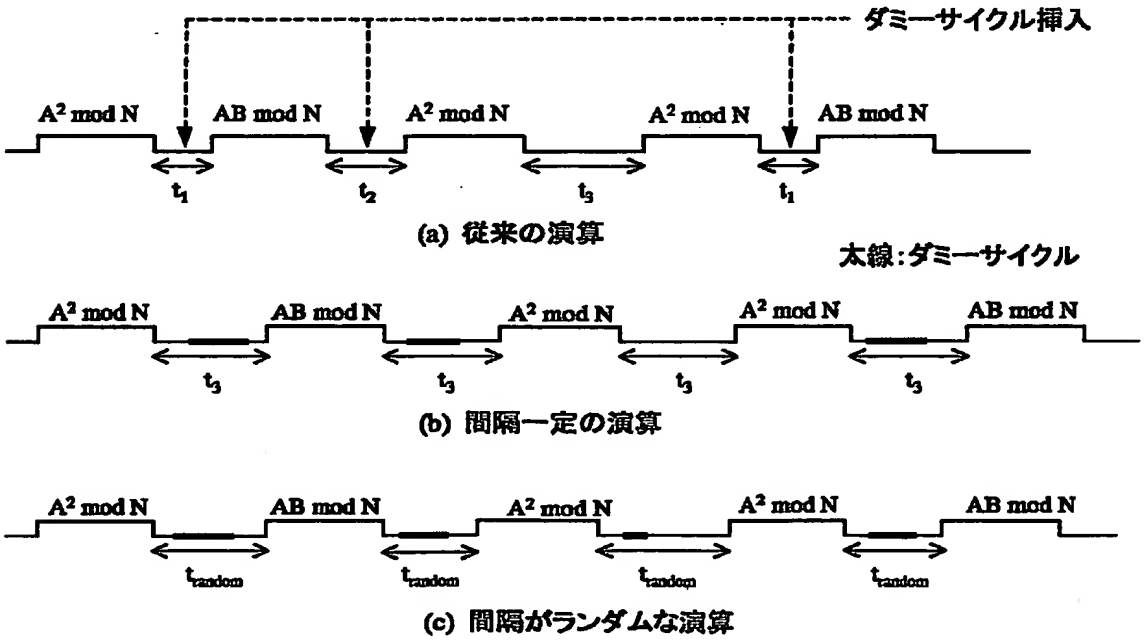
【図１３】


(a)
(b)
コプロ演算
オーバーフロー処理
(ダミー)
コプロ演算
オーバーフロー
処理
コプロ演算
オーバーフロー
処理
対策前
対策後

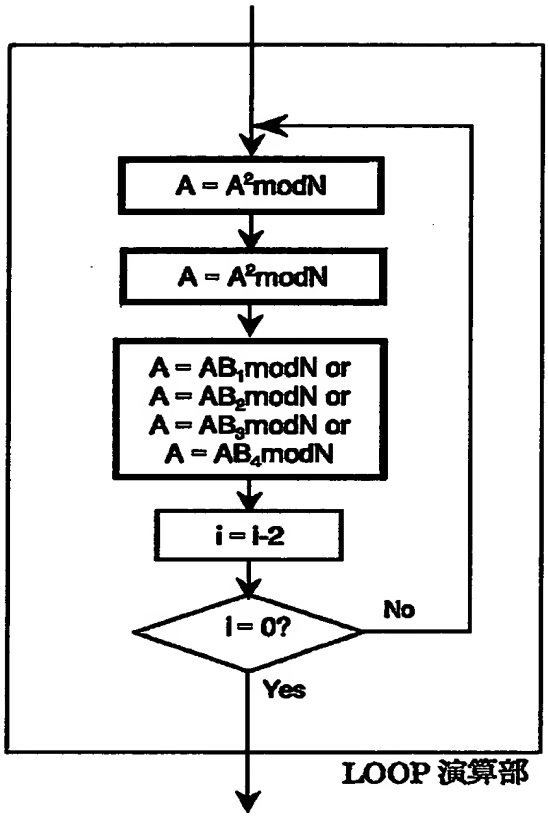
【図１４】


制御論理
R/Wバッファ
又は
レジスタブロック
コプロオーバーフロー
演算器
ライトストローブ
ダミーオーバーフロー

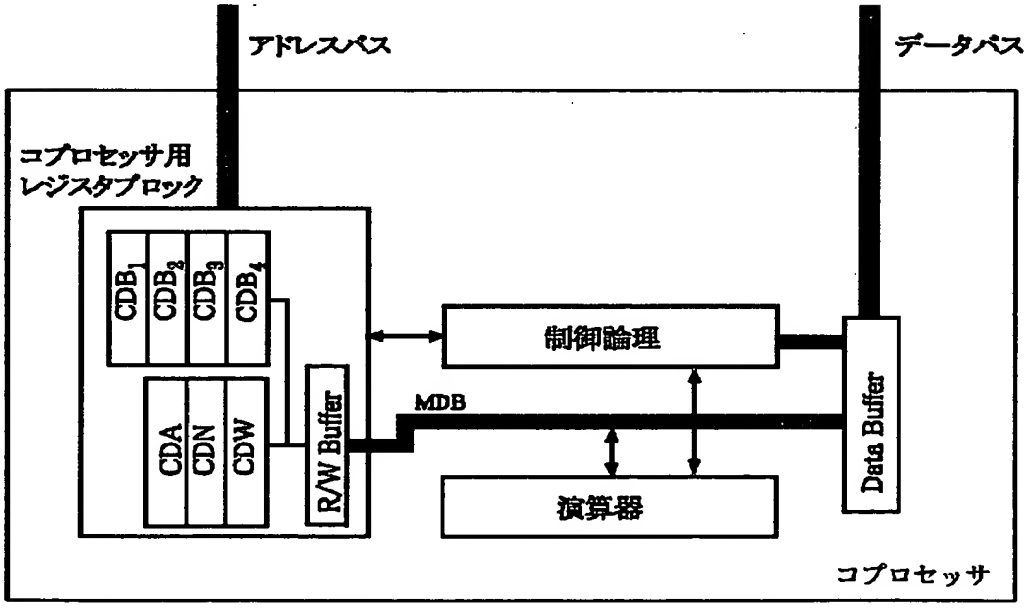
【図１５】


ダミーサイクル挿入
A² mod N
AB mod N
A² mod N
A² mod N
AB mod N
t1
t2
t3
t1
(a) 従来の演算
太線:ダミーサイクル
A² mod N
AB mod N
A² mod N
A² mod N
AB mod N
t3
t3
t3
t3
(b) 間隔一定の演算
A² mod N
AB mod N
A² mod N
A² mod N
AB mod N
trandom
trandom
trandom
trandom
(c) 間隔がランダムな演算

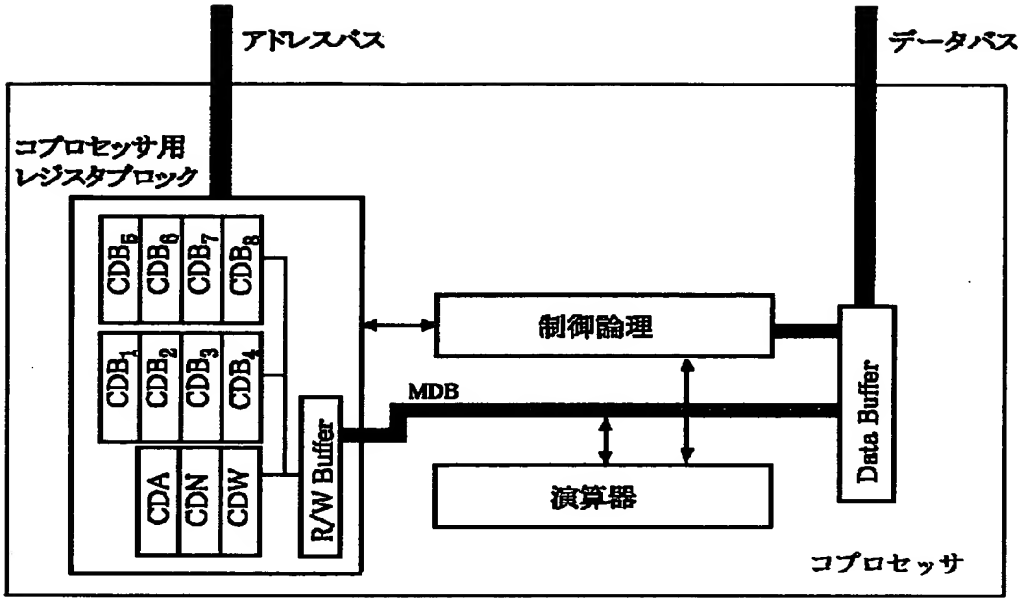
【図１６】


A = A²modN
A = A²modN
A = AB1modN or
A = AB2modN or
A = AB3modN or
A = AB4modN
i = i-2
i = 0?
No
Yes
LOOP 演算部

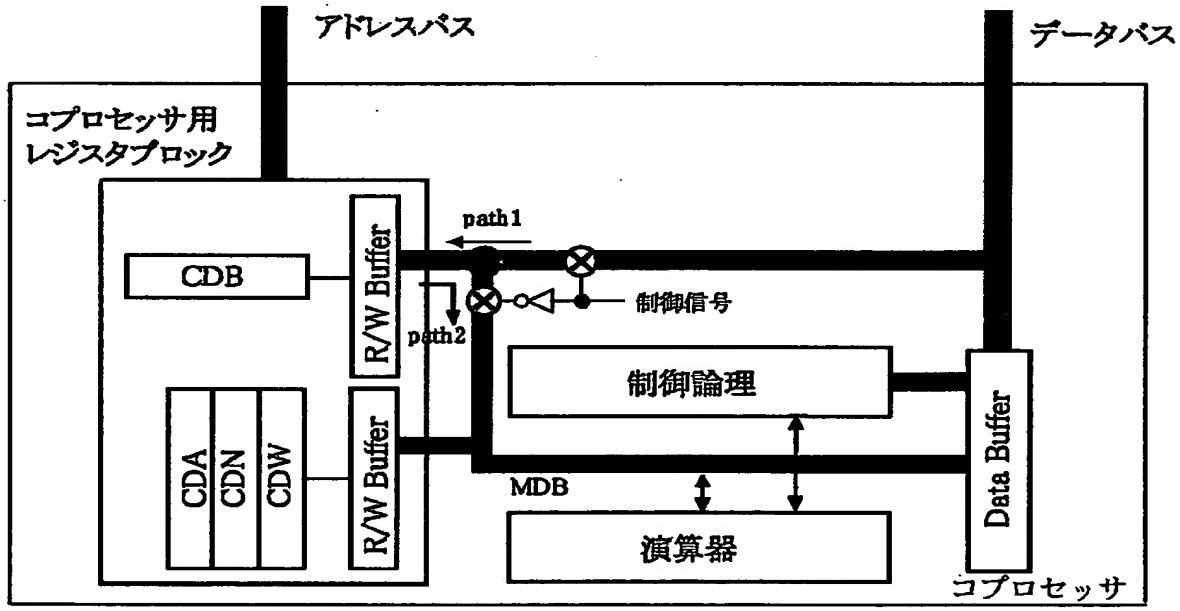
【図１７】


アドレスバス
データバス
コプロセッサ用
レジスタブロック
CDB1
CDB2
CDB3
CDB4
CDA
CDN
CDW
R/W Buffer
制御論理
MDB
演算器
Data Buffer
コプロセッサ

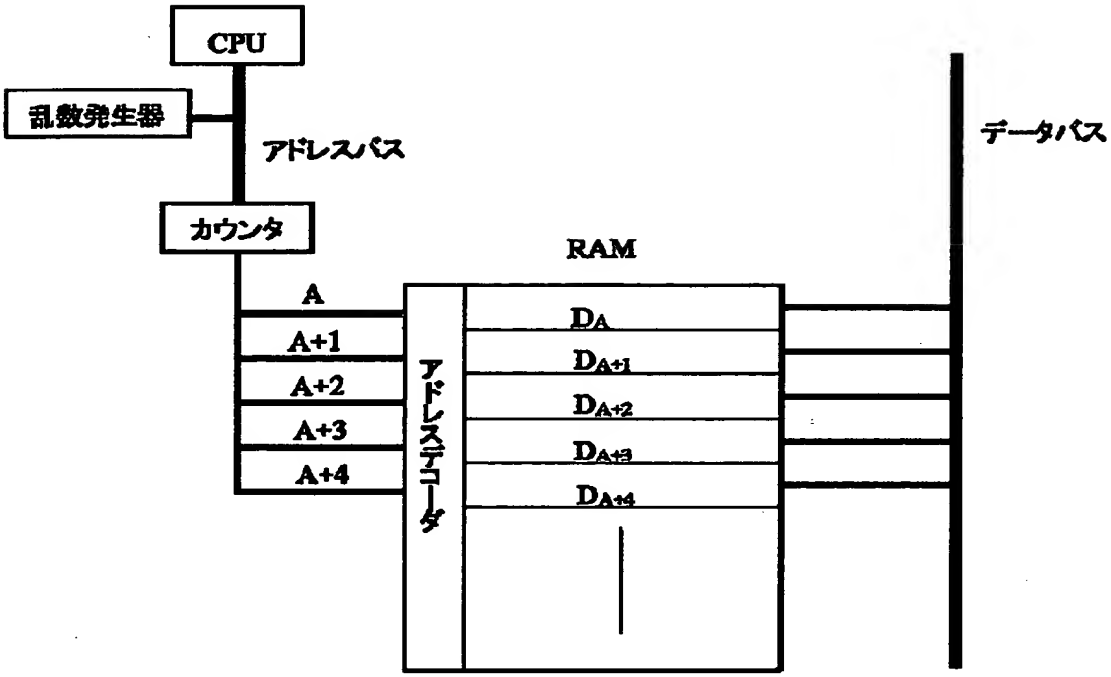
【図１８】


アドレスバス
データバス
コプロセッサ用
レジスタブロック
CDB5
CDB6
CDB7
CDB8
CDB1
CDB2
CDB3
CDB4
CDA
CDN
CDW
R/W Buffer
制御論理
MDB
演算器
Data Buffer
コプロセッサ

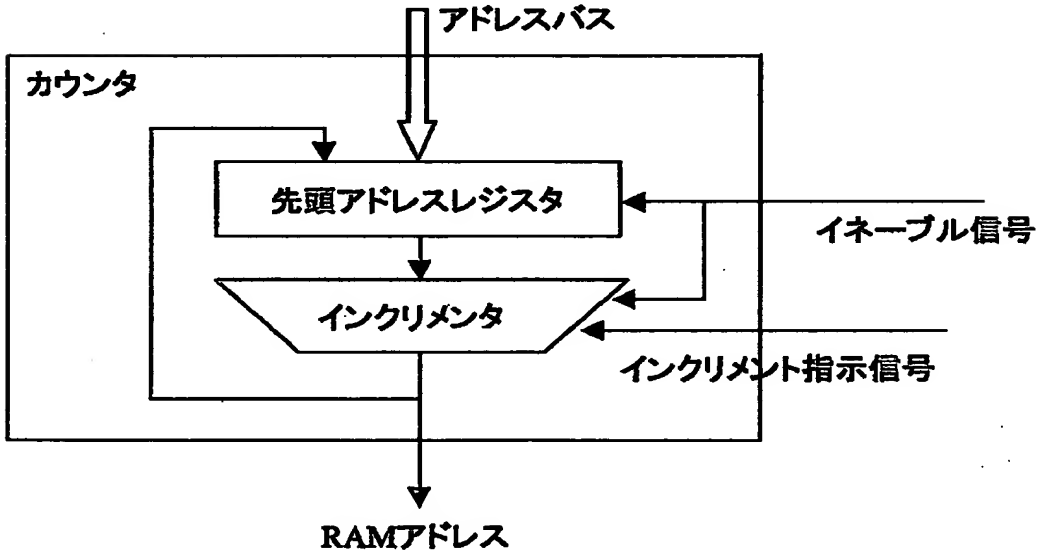
【図１９】


アドレスバス
データバス
コプロセッサ用
レジスタブロック
CDB
R/W Buffer
path1
path2
制御信号
制御論理
CDA
CDN
CDW
R/W Buffer
MDB
演算器
Data Buffer
コプロセッサ

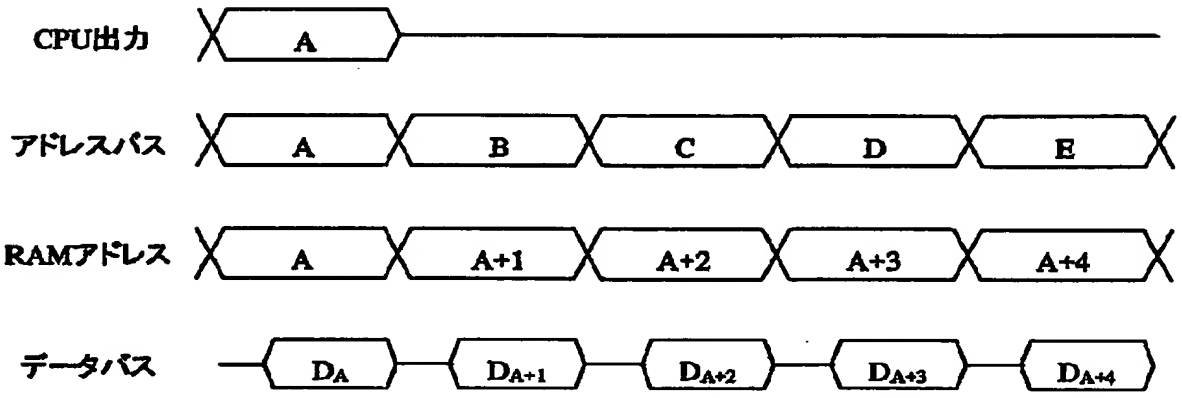
【図２０】


CPU
乱数発生器
アドレスバス
カウンタ
RAM
データバス
A
A+1
A+2
A+3
A+4
アドレスデコーダ
DA
DA+1
DA+2
DA+3
DA+4

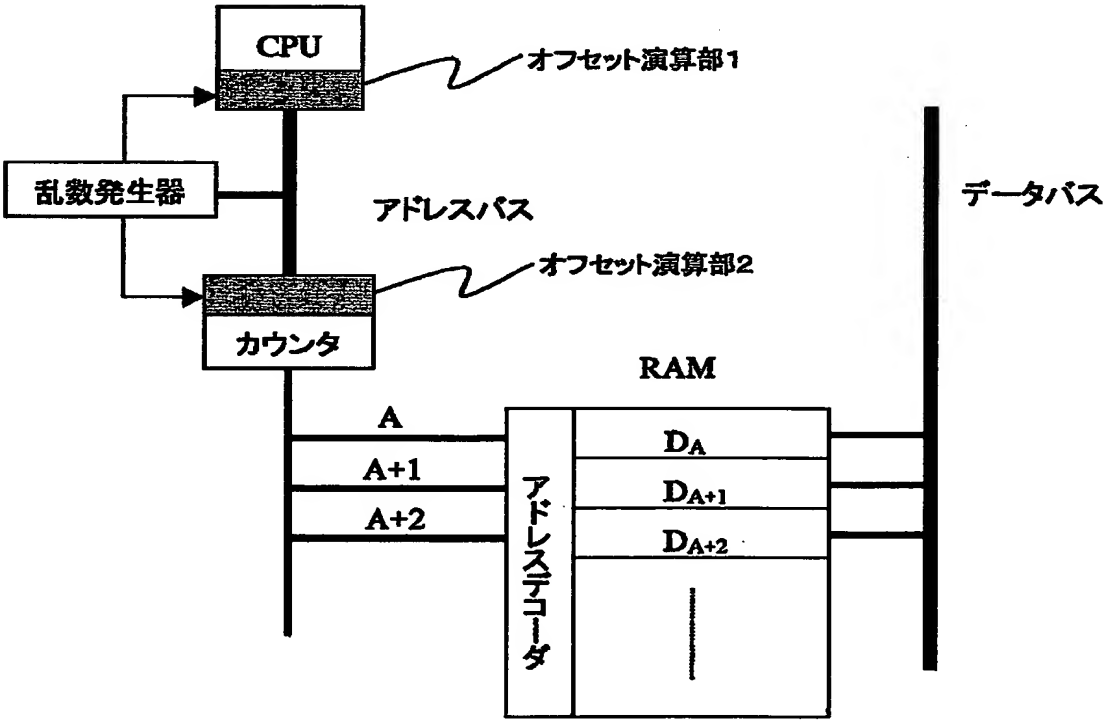
【図２１】


アドレスバス
カウンタ
先頭アドレスレジスタ
イネーブル信号
インクリメンタ
インクリメント指示信号
RAMアドレス

【図２２】

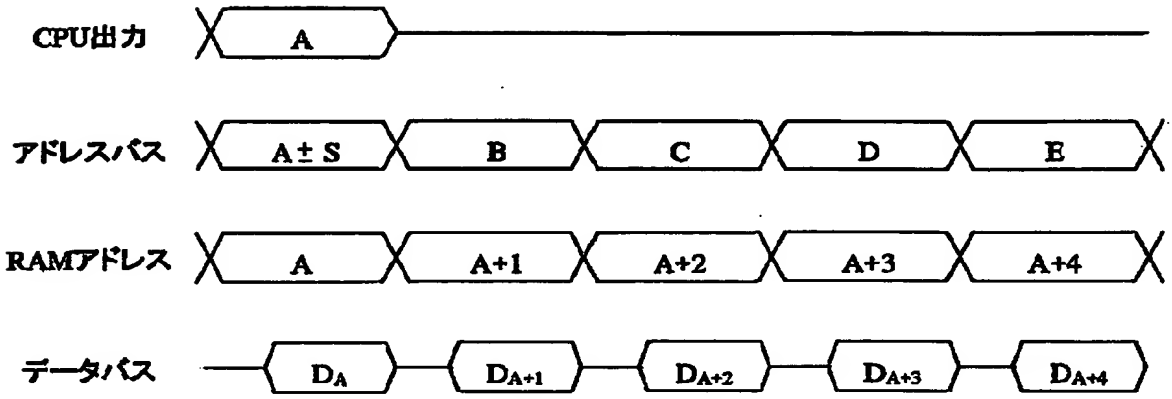


【図２３】

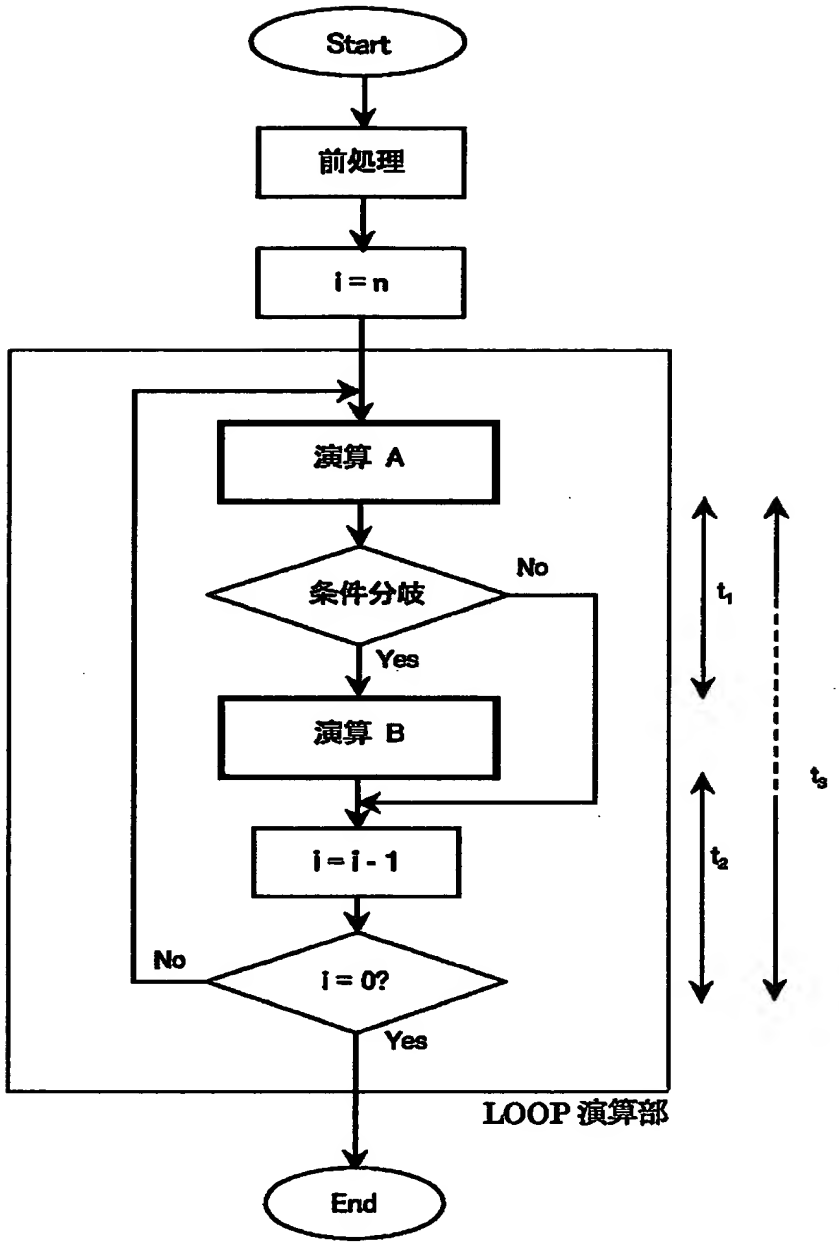


出証特２０００－３１１１６６９

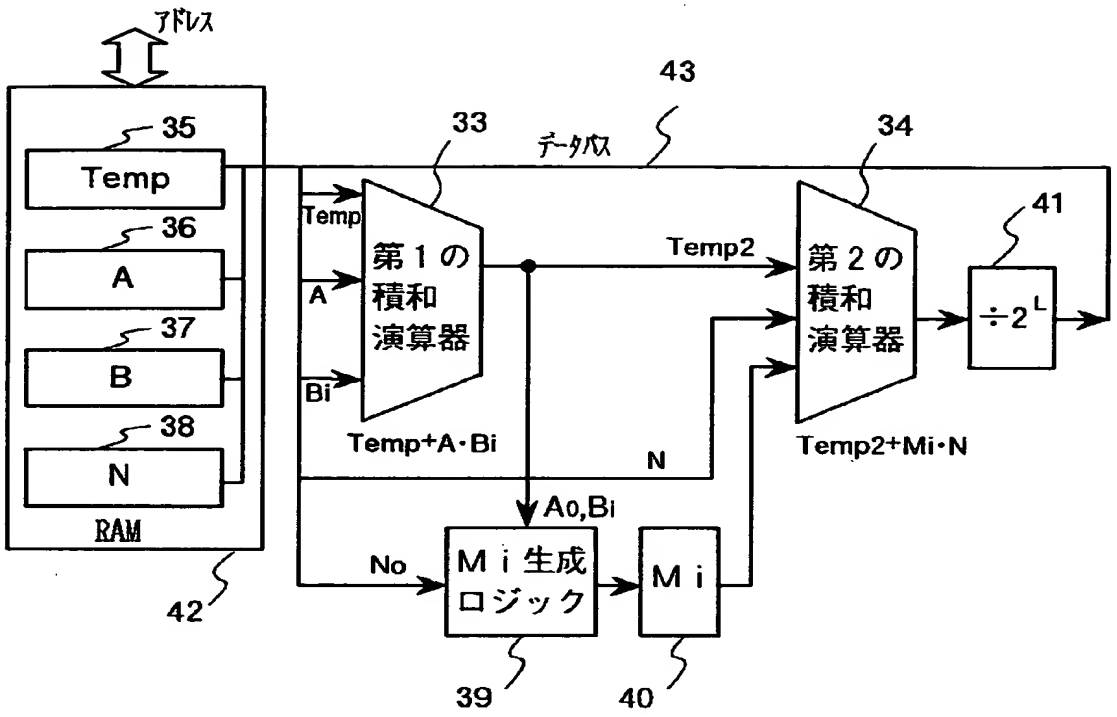
【図２４】


CPU出力
A
アドレスバス
A±S
B
C
D
E
RAMアドレス
A
A+1
A+2
A+3
A+4
データバス
DA
DA+1
DA+2
DA+3
DA+4

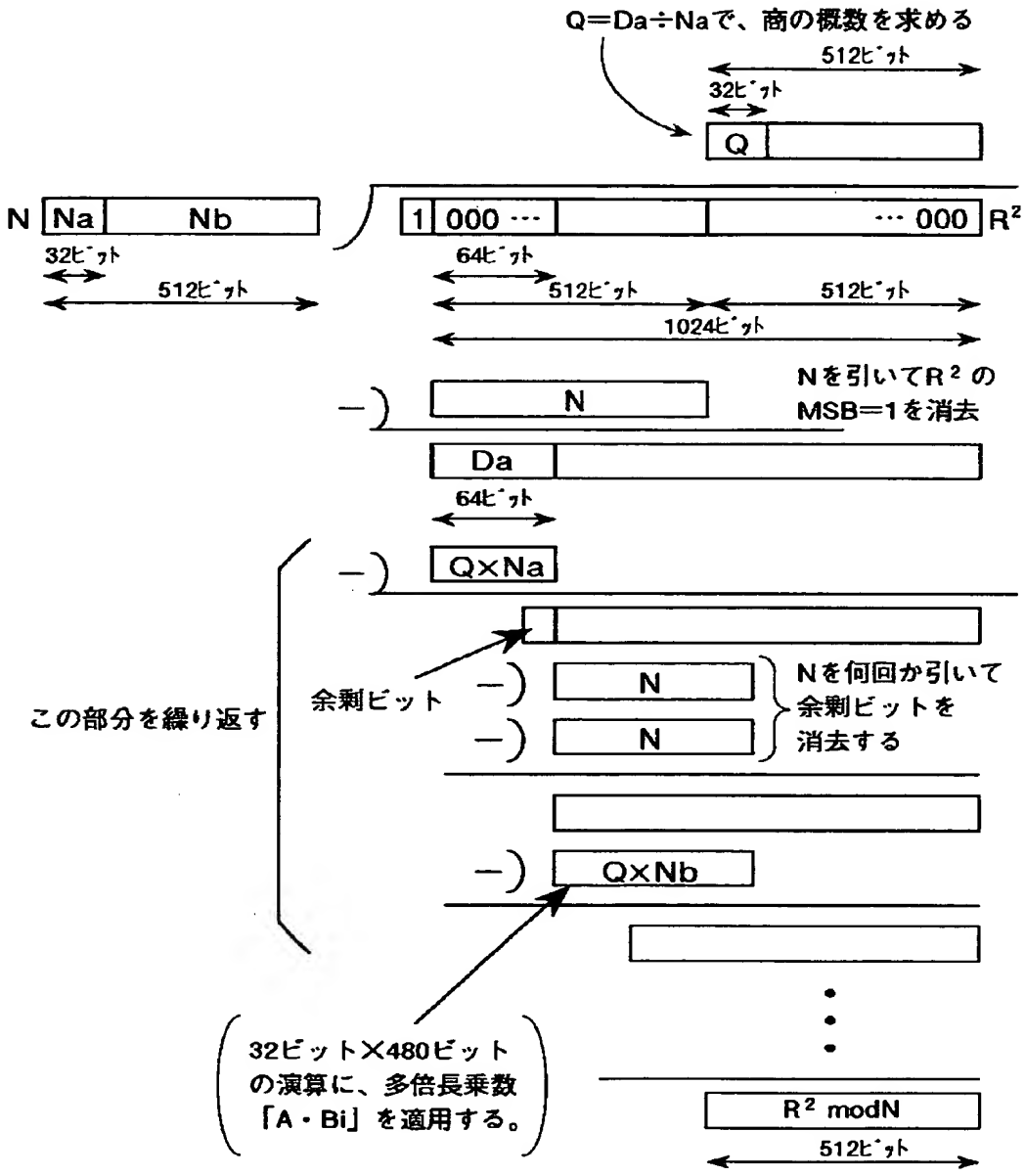
【図２５】


Start
前処理
i = n
演算 A
条件分岐
No
Yes
演算 B
i = i - 1
No
i = 0?
Yes
LOOP 演算部
End
t1
t2
t3

【図２６】

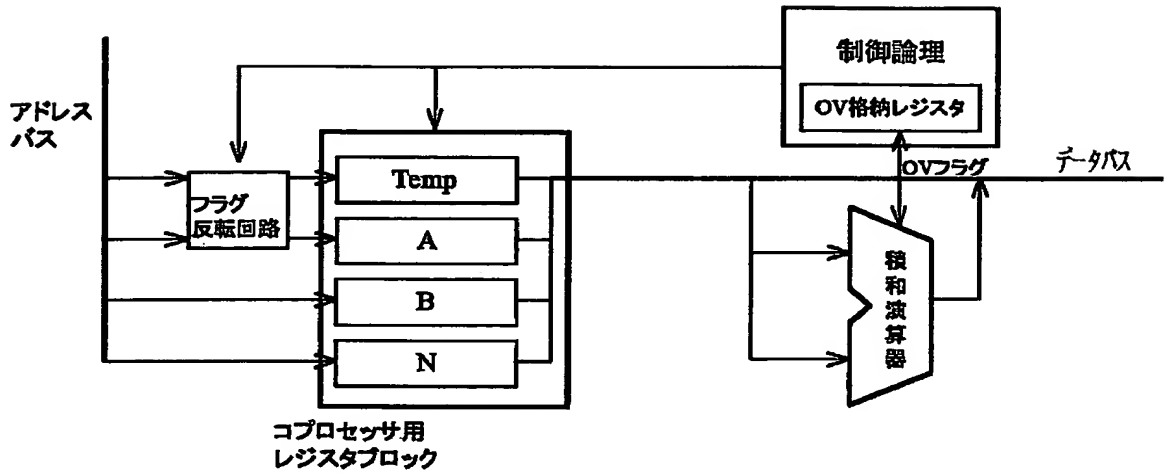

アドレス
43
35
Temp
36
A
37
B
38
N
RAM
42
33
データバス
34
Temp
A
Bi
第１の
積和
演算器
Temp+A･Bi
Temp2
第２の
積和
演算器
Temp2+Mi･N
41
÷2^L
N
A0,Bi
No
Ｍｉ生成
ロジック
Ｍｉ
39
40

【図２７】

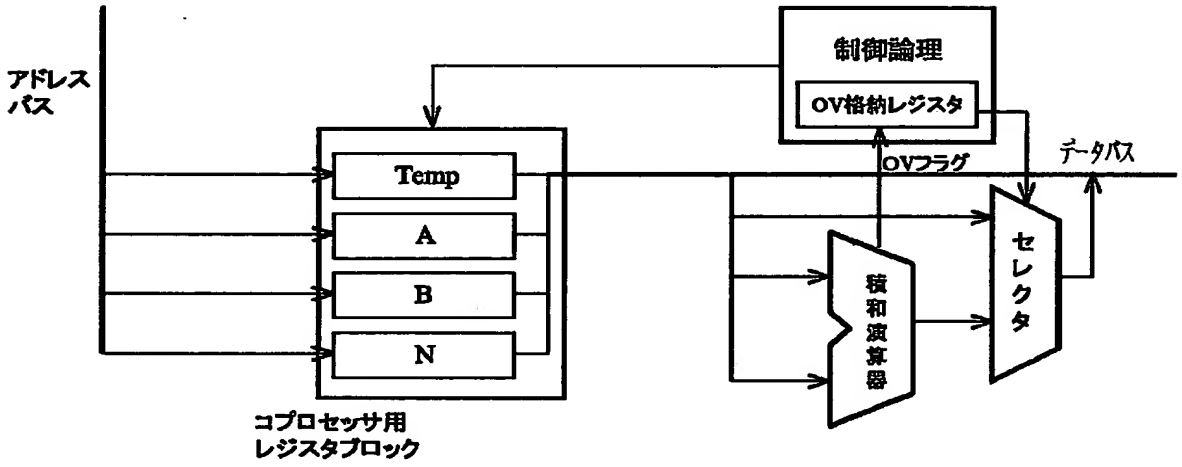


R²modNの計算概念図

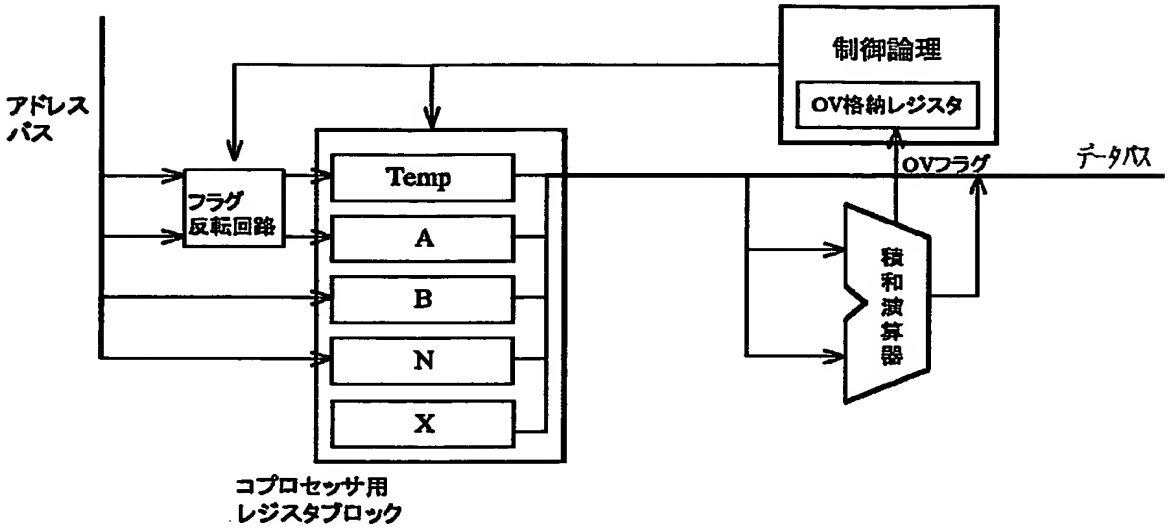
【図２８】


制御論理
OV格納レジスタ
アドレス
バス
フラグ
反転回路
Temp
A
B
N
OVフラグ
データバス
積和演算器
コプロセッサ用
レジスタブロック

【図２９】


制御論理
OV格納レジスタ
アドレス
バス
Temp
A
B
N
OVフラグ
データバス
積和演算器
セレクタ
コプロセッサ用
レジスタブロック

【図３０】


制御論理
OV格納レジスタ
アドレス
バス
フラグ
反転回路
Temp
A
B
N
X
OVフラグ
データバス
積和演算器
コプロセッサ用
レジスタブロック

【書類名】　　要約書

【要約】

【課題】　機密保護の強化及び機密保護のための信号処理の高速化とその強化を実現したＩＣカードとマイクロコンピュータを提供する。

【解決手段】　外部端子がリードライト装置と電気的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ったデータの入出力動作を含むＩＣカードにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにする。暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにする。

【選択図】　　図４

出　願　人　履　歴　情　報

識別番号　　〔０００００５１０８〕

１．変更年月日　　１９９０年　８月３１日
［変更理由］　新規登録
住　所　東京都千代田区神田駿河台４丁目６番地
氏　名　株式会社日立製作所

出　願　人　履　歴　情　報

識別番号　　〔０００２３３１６９〕

１．変更年月日　　１９９８年　４月　３日
［変更理由］　名称変更
住　所　東京都小平市上水本町５丁目２２番１号
氏　名　株式会社日立超エル・エス・アイ・システムズ